HEARTFAID

D15 – Functional specifications of Data Processing and Decision Support Services

Submission date: 09/03/2007
Due date of document: 01/02/2007

HEARTFAID

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

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

Consortium

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

D15 – Functional specifications of Data Processing and Decision Support Services

Document summary	
Document title:	D15 – Functional specifications of Data Processing and Decision Support Services
Document classification:	Derivable D15
Dissemination level:	CO
Submission date:	09 March 2007
Due date:	01 February 2007
Authors:	Sara Colantonio, Massimo Martinelli, Davide Moroni, Patrizia Asirelli, Umberto Barcaro, Edoardo Bozzi, Massimo Chimenti, Gabriele Pieri, Ovidio Salvetti– CNR Franco Chiarugi, Vangelis Sakkalis – FORTH Antonio Candelieri, Domenico Conforti, Domenico Costanzo, Rosita Guido, Vincenzo Lagani – UNICAL Francesco Perticone, Angela Sciacqua – UNICZ Dragan Gamberger, Marin Prcela - RBI
Work package:	WP5 – Data Processing and Decision Support Services
Report version:	1.0

Short description
This deliverable analyses the methodological and technical requirements and specifications inherent in the integration of computational modelling, knowledge discovery, signal and image processing, as well domain medical understanding, in order to design and develop an effective and reliable *Clinical Decision Support System*, the *HEARTFAID CDSS*, corresponding to the core of the HEARTFAID *intelligence*. Emphasis is given to the definition of innovative approaches for cardiac signal and image processing and for robust and reliable reasoning, based on Machine Learning and inference methodologies on declarative and procedural domain knowledge.

Change record		
Version number	Changes	Release date
0.1	Template	12/01/2006
0.2	Preliminary draft	05/02/2007

0.3	First draft	16/02/2007
0.4	Advanced draft	23/02/2007
0.5	Final draft	07/03/2007
1.0	Final version	09/03/2007

Table of contents

Executive Summary

HEARTFAID aims at defining efficient and effective health care delivery organization and management models for an "optimal" patients' management in the field of cardiovascular diseases. An informative and decision support platform of services will be devised for improving all the processes related to diagnosis, prognosis, treatment and personalization of health care of the Heart Failure (HF) in elderly population.
To this end, innovative results on computational modelling, knowledge discovery methodologies, visualization and imaging techniques, and the medical knowledge of the relevant domain will be opportunely integrated to design and develop an effective and reliable *Clinical Decision Support System (CDSS)*: the *HEARTFAID CDSS*, corresponding to the core of HEARTFAID *intelligence*. This system will be able to process clinical knowledge and patient-related information, intelligently filtered, processed and presented at appropriate times, to enhance patient care. It will be devised as a service of the HEARTFAID platform for providing an effective support to the daily practice of the clinicians, by implementing adequate data processing algorithms, by providing guidelines to medical protocols as well as access to the knowledge base, by sending alarms in case of critical situations, and by supplying diagnostic suggestions.

Two peculiar issues are involved in the development of the HEARTFAID CDSS for supporting medical decision making, i.e.

- innovative approaches for biomedical signal and image processing;
- robust and reliable *reasoning* approaches, based on *Machine Learning* and inference methodologies on declarative and procedural domain knowledge.

The first, fundamental step towards the design and development of the HEARTFAID CDSS consists in analysing the requirements and specifying the functional details of both data processing and decision support services. These topics are the focus of this deliverable, which collects the results of Task T5.1, "*Identification of representation features for signals and images processing*", and Task T5.3, "*Requirements and functional specification of the Decision Support System*", of Work Package WP 5.

A deep investigation into issues inherent in (i) data typologies and their characterization, (ii) reasoning methodologies, (iii) HEARTFAID CDSS logical-functional vision, and (iv) current available technologies are herein provided, with the final aim of supplying the guidelines for the development of the HEARTFAID data processing and decision support services.

The document layout has two main "souls", addressing, from one side, all the methodological and technological aspects of signal and images processing, and from the other, all the topics related to decision support. Moreover, it is ideally

divided into two parts: in the first part (Chapters 2, 3, and 4) the discussion consists of a revision of the State of the Art; whereas in the second part (Chapters 5, 6, 7, and 8) the attention is focused on the HEARTFAID specificity, regarding the final goal of the deliverable.

More in detail, the document is organized as follows.
At first, an introduction supplies an overview of the problems to face for reaching the HEARTFAID Grand Vision about the intelligent support to the medical decision making.

In Chapter 2, the available diagnostic resources for signals and images acquisition are introduced, analysing their peculiarities, relevance to the HEARTFAID clinical domain, and open-problems. An overview of the principles of signal and image processing and of the mathematical models for features extraction is also provided.

Afterwards, in the third chapter, a careful survey of the methodological foundations of decision support systems (DSS) is reported, discussing the main aspects of decision theory and typical DSS models and structures. According to HEARTFAID definition as a knowledge-based platform of services, medical domain knowledge plays a fundamental role in HEARTFAID decision support development. Thus, particular attention is reserved to the category of *Expert Systems* or *Knowledge-based* DSS, and related issues, such as knowledge representation and inference engine modelling. Clinical applications and topics inherent to CDSS, e.g. guidelines modelling, are widely addressed as well, for better understanding the design implications involved.
The chapter ends with a discussion about the important issues regarding DSS design and success factors, which should be taken into account when devising the HEARTFAID CDSS.

Chapter 4 reports a critical description and analysis of the available up-to-date technologies for both data processing and decision support services.

In Chapter 5, the biomedical parameters, relevant to HF and selected by the medical partners, are critically reviewed in the perspective of data processing. From this analysis, the most relevant problem in data processing w.r.t. HEARTFAID platform is selected and the functional specifications of the planned signals and image processing are described.

The HEARTFAID clinical decision making problems are deeply investigated in Chapter 6, in order to identify the overall objectives of the HEARTFAID CDSS, i.e. what it should accomplish and why, specifying the element of the knowledge to be formalized. The system *requirements* are defined responding to HF clinicians' needs, as also stated in the deliverable D5, in order to devise an efficient and usable system for supporting the medical personnel in their daily activity.

Furthermore, general characteristics related to system efficiency, portability and adaptability are highlighted and discussed.

Chapter 7 concerns the specification of the general capabilities that the HEARTFAID CDSS should have to satisfy the requirements. The system *functionalities* are described by the characterization and elicitation of the key concepts and relations among the knowledge elements, and by identifying the processing components that compose its architecture. For each of them, specific functional specifications are provided. Moreover, the advantages of the designed architecture are also underlined, together with some considerations about its optimization and extension.
Finally, some guidelines concerning the HEARTFAID CDSS development are supplied in the last chapter.

1. Introduction

The application of computers in health care is already an ‘‘old’’ pursuit: efforts to automate aspects of health care began more than 45 years ago, in the early 1960s, and, since then, a number of computer-based clinical decision aids have been developed, addressing, in many cases, only simple types of clinical decisions (e.g., recognizing that a laboratory test result is out of normal range, or that a medication being ordered has a dangerous interaction with another one).
In the last years, there has been a paradigm shift in clinical information systems. Formerly, the foci of development were the electronic medical record, information retrieval and reporting, scheduling, and various communications functions, as well as financial and billing applications. Whereas, over the past decade, emphasis is shifting increasingly to cost-effectiveness, error prevention, safety, and improvement of health care quality (Greenes 2003).
Actually, health care practitioners have to continually face a wide range of challenges, trying to make difficult diagnoses, avoid errors, ensure highest quality, maximize efficacy, and save money, all at the same time. Nevertheless, there are increasing amounts of medical knowledge, of new information about disease causes, of diagnostic and therapeutic options, and mountains of patient data from ever-expanding numbers of diagnostic tests and imaging studies. In addition, the amazing possibilities offered by high-tech instrumentations often impress patients, feeding their expectations and making them putting pressure on clinicians to perform all relevant diagnostic tests and to prescribe the latest, most expensive treatments. As a consequence, the array of choices, the tradeoffs, and other mitigating factors that affect clinical decisions are becoming more and more complex and requiring more detailed knowledge than ever before.
All these conditions are creating a fertile environment for developing and using computerized applications, which, with their speed, vast memories, and stored knowledge, can surely provide patient-specific, point-of-care decision support.
Evidence proving this conclusion is a report released by the American Medical Informatics Association (AMIA), titled “*A Roadmap for National Action on Clinical Decision Support*”, issued last June 2006[1]. *Clinical decision support* (CDS) is described as a collection of a number of approaches for providing clinicians, care staff, patients or other individuals with knowledge and person-specific information, intelligently filtered or presented at appropriate times, that can improve decision making, prevent errors, and enhance health and health care (Osheroff *et al*. 2004; W1).
A plethora of *Clinical Decision Support Systems* (CDSS) has already evolved with different platforms and architectures, encompassing a variety of tools and interventions, such as computerized alerts and reminders, clinical guidelines, provider order entry, diagnostic support, clinical result interpretation, adverse event monitoring, shared patient-doctor decision-making. In particular, a great

[1] This report is just one of several publications suggesting the adoption of computerized applications for supporting routine clinical practice (Greenes 2003, W2, Kawamoto *et al*. 2005).

effort of research is being devoted to computer-based clinical guidelines and protocols, since it has been demonstrated that their compliance leads to a reduction of practice variability and patient care costs, while improving quality and safety in healthcare.
However, the implementation of effective CDSS is a challenging task involving interactions between technologies and organisations, and there are no easy solutions to guarantee success or to avoid failure. The imperative requirement is incorporating executable high-quality, evidence-based medical knowledge and point-of-care decision support. Tools for acquiring, encoding, and managing the underlying knowledge bases are then essential, since a CDSS is as effective as its underlying knowledge base, which changes rapidly as medical science evolves. Moreover, during the process of care, the knowledge should be accessible for revision, inspection, and update to domain experts when needed.
In this context, the Roadmap identifies three pillars that are needed to support widespread and optimal use of CDSS to improve the quality of health and care delivery:

i. Make the best knowledge readily available when needed, by building highly practical formats and services for representing, collecting, organizing, and distributing clinical knowledge and CDS interventions;
ii. Foster increased adoption and effective use, by organizing and publishing best strategies for improving CDS system design, usability, and implementation, as well as strategies for addressing legal and financial barriers;
iii. Continuously improve CDS interventions and health-related knowledge, by developing systematic methods for sharing CDS experience and for continuous improvement based on feedback, experience, and data that are easy to aggregate, assess, and apply, leveraging electronic health records to enhance clinical knowledge.

The HEARTFAID Platform (HFP), as a complex clinical information management system, able to offer knowledge-based decision support services within the Heart Failure (HF) clinical management domain, is founded on these pillars. Actually, the HFP is specifically characterized by the following innovative peculiarities:

- integration of several approaches for coding the relevant medical knowledge and extracting new knowledge: a knowledge based approach (*deductive knowledge*) for coding the clinical guidelines and the clinical best practice; a *data mining* approach (*inductive knowledge*) for extracting new knowledge from the practical clinical experience represented by suitable sets of cases (*Knowledge Discovery in Database* – KDD);
- clinical decision support, based on pattern recognition in historical data, computational analysis and inferences on patients' clinical data, for facing all the clinical management of HF patient: diagnosis, prognosis, therapy planning.

- integrated services for healthcare professionals, including easy access to heterogeneous patients' data and to formalized knowledge, patient telemonitoring, signal and image processing, alert and alarm system.

The strategic impact of the HFP in terms of healthcare improvements is twofold: the benefits assured concern both patients' quality of life and clinicians' daily activity. In fact, within the HEARTFAID environment, an innovative health care management program (HFCMP) will be provided for HF patients (Deliverable D8). HFP contributes to the HFCMP definition and, then, to the improvement of indices of health-related quality of life by:

- making the therapeutic treatment more effective, appropriate and personalized;
- realizing a real-time monitoring and assistance of the HF patient, in order to reduce the risk for adverse events.

This in turn will assure a reduction of the overall economic and social costs of medical care, since the early detection of HF related signs and symptoms of patient's *decompensation* allows decreasing the frequency of hospital admissions.
The HFCMP will be *patient centric*, organized by the suitable involvement and competence integration of different health care environments and operators, i.e.

- Home Care environment: the patient himself and his family are considered as active users;
- Primary Care environment: interaction between the patient and the practitioner;
- Ambulatory with internist physicians and professional nurses working on the territory;
- Hospital including cardiology, geriatric and internist medicine wards**.**

The main objectives of a HF therapy can be summarised as follows: slow down the progression of the disease, alleviate symptoms, and minimize risk factors.

On the other hand, clinicians and care operators will be supported at decisional levels by

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

Such facilities will improve the effectiveness and efficiency of HF clinical practise, and promote a better knowledge and understanding of HF.

In complete accordance to what stated in the Roadmap, the core components of the HFP are represented by:

- the formalized clinical knowledge, consisting of the pre-existing guidelines and experts' know-how procedures, and the new elicited knowledge discovered by KDD processes;

- the clinical decision support services, consisting of robust and reliable *reasoning* approaches, based on computational models of *Machine Learning*, inference methodologies on declarative and procedural domain knowledge, and innovative methods for data processing.

The combination of these two components, which can be respectively considered as the *heart* and the *brain* of the HFP, constitutes the *HEARTFAID Clinical Decision Support System* (HEARTFAID CDSS). In particular, a fundamental functionality of this system is represented by signals and images processing, which is worth of a specific independent discussion when describing the HEARTFAID CDSS.

All components will be integrated into the HFP using interoperability tools, in order to devise a multi-level heterogeneous and distributed architecture where each level has different responsibilities and provides integrated functionalities to the adjacent levels.

In order to clarify the HFP integration, the HEARTFAID architecture levels (Figure 1.1), introduced in the Description of Work, are herein described:

- Biomedical Data level (Biomedical data collection and transmission): it is the lower level, concerned with the heart failure related biomedical data. This part of the platform 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 biomedical data level interacts with the middleware level, which is responsible for the exchange of data among the platform modules and it is in charge of guaranteeing the interoperability both inside the platform 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 among the internal modules of the platform and between the platform and the external applications are compliant with the standards for clinical data representation and communication.
- Knowledge level (Data preparation, Knowledge Discovery in Database and Ontologies): it 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 project repository by using innovative knowledge discovery processes.
- Decision Support level (Decision Support System and Signals/Images processing): it provides an effective support to the daily practice of the clinicians in the field of cardiovascular diseases prevention by implementing adequate data processing algorithms, providing guidelines to medical protocols as well as access to the knowledge base, alarms in case of critical situations and diagnostic suggestions.
- End-users level: it 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.

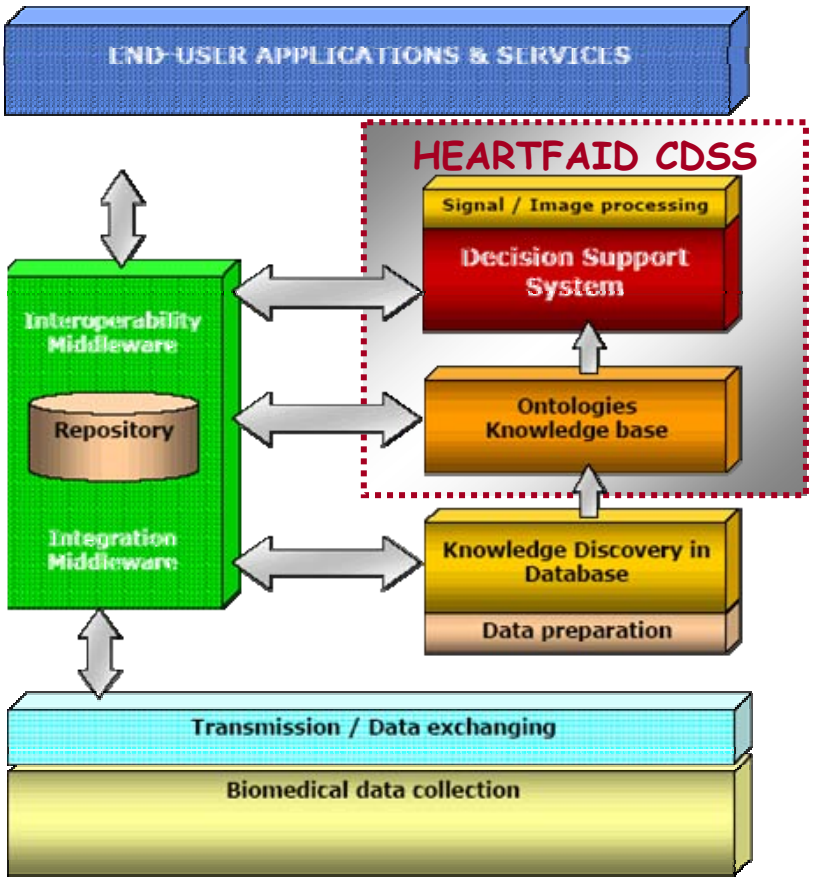


Figure 1.1. *The HEARTFAID Architecture, with highlighted the components of the HEARTFAID CDSS*

The described multi-level structure allows fulfilling requirements that are becoming more and more imperative for reliable CDSS:

- standardization, which allows sharing and reusing knowledge and decision support applications;
- flexibility, corresponding to the possibility of incorporating information from diverse sources;
- adaptability, that is the ability to adapt to varied practice settings and to reflect changes occurring in the medical domain;
- extensibility, which permits easy additions of new knowledge and services.

The scope of this document is a comprehensive analysis and discussion of the requirements and functional specifications of the HEARTFAID CDSS. This entails an insight into the characteristics of the HF decisional problem to support,

both in term of data and knowledge involved and the reasoning mechanisms followed by the clinical operators. Moreover, being a component of a complex, multi-level infrastructure, the interaction with the other components will also be a matter of investigation.

An overview on the general activity of the HFP can be assumed as starting point for the insightful analysis, followed by a more detailed perspective on specific aspects to face for data processing and decision support delivery. These topics are the focus of the following sections.

1.1 The HEARTFAID General Activity Cycle

The general activity of the HEARTFAID platform can be synthesized as an iterative *cycle*, consisting of the following steps:

- *Measurement*
- *Analysis*
- *Decision*
- *Action*

Using a more schematic formalism, we can describe the general workflow of the platform with the block diagram of Figure 1.2.

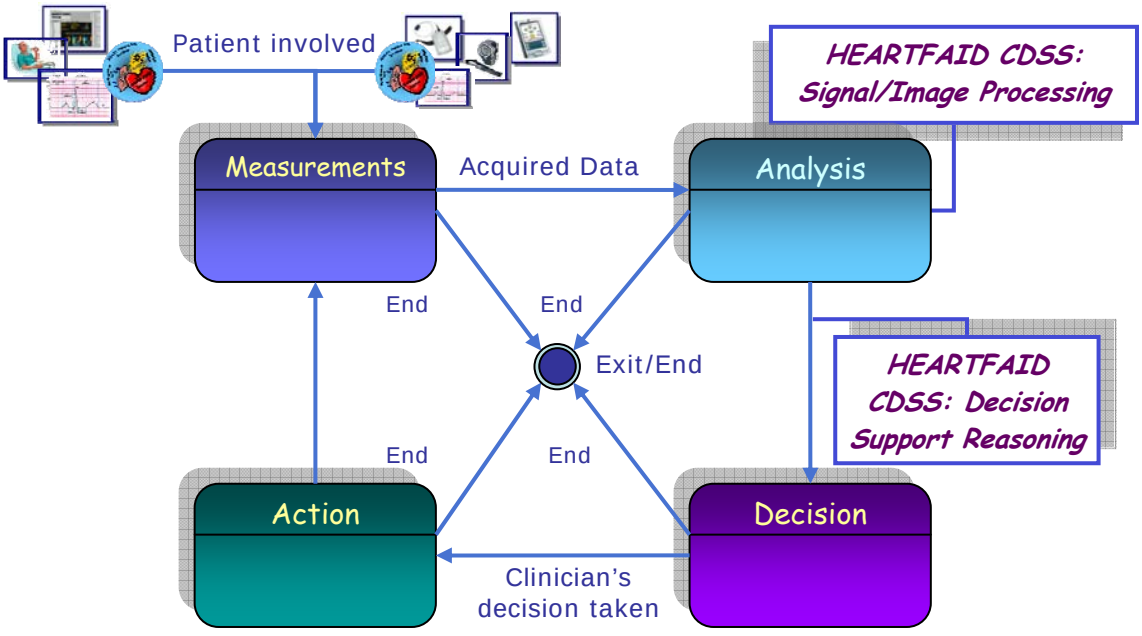


Figure 1.2. *The HEARTFAID Cycle: the interventions of the HEARTFAID CDSS are highlighted.*

Within the HFP, such cycle will be instantiated and adapted to each of the three typical HEARTFAID patients' scenarios, i.e. hospital environment; home care; patient on the move.
In order to design the main functionalities of the platform, it will be important to define, for each step of the iterative cycle, the following information:

- data to be managed, i.e. which data, which devices and which clinical acquisition protocol;
- relations among data: data can be directly measured and obtainable from the devices or can be evaluated after analysis of the raw data acquired by the medical devices. Thus, we will define two kinds of data: directly measured data and calculated data. Calculated data are all measurements that are not directly provided by the medical device itself, but need some additional analysis (with ad hoc developed SW) for their computation. These data should not be confused with the historical data used in the KDD process to discover novel know-how. The DSS analyses new acquired data according to the rules coded in the knowledge base in order to identify relevant situations. The KDD processes historical data (i.e. belonging to more patients) to identify recurring patterns of interest for the specific cardiovascular domain.
- flows of information.

This will be the strategy adopted within this document to detail the functional details of the decision support services.
For the first step of the cycle, *Measurements*, consists in the collection of all the data relevant for assessing the patient's state, such as patient's symptoms, anamnesis, diagnostic examination results (i.e. diagnostic signals and images), vital signs parameters from telemonitoring, and so forth. We need to specify accurately what will be the data available and that need to be managed, as well as what are the correlations among these data.
The second step, *Analysis*, can be considered from two points of view:

1. From one side it will be necessary to support the daily activity of the medical experts by automating operations that are currently performed. In other words, the platform should be able to process the available data according to the explicit knowledge of the experts and the know-how available in the specific HF field;
2. From the other side, the HFP should analyse the available data by means of both standard and advanced mathematical methods in order to extract new information that can be successively used in KDD and statistical process to derive new information, to infer new knowledge or to identify correlations among data.

The intervention of the HEARTFAID CDSS will be provided at this step for data processing, when dealing with diagnostic signals and images evaluation.

The *Decision* step corresponds to the daily decision activity of the clinicians by exploiting the know-how derived from guidelines, protocols, domain knowledge. The HEARTFAID CDSS has the precise goal to support the process of moving from the "analysis of data" to the "decision making". Inference and computational reasoning processes, on knowledge bases and source data, will be employed to this end.
The last step, *Action*, is a direct consequence of decision making. In fact, according to the measurements acquired from the specific patient, the knowledge derived by the DSS and the decisions taken by the experts, changes will be likely applied to patient's healthcare program. These changes can be considered of two types:

- changes to the therapy and patient's habits;
- changes to the HFP in terms of configuration and functionalities offered to the patient.

1.2 Data Processing in the HEARTFAID Platform

Analysis of the biomedical signals and images relevant to HF and the definition of representation features is a key-point of various levels of HFP. Regarding the routine clinical practice environment, there is the need to provide well-established and defined parameters extracted from the data. The assessment of their independence from the chosen modality or even the calculation method is of crucial importance, since these measurements will be eventually one of the inputs of the HEARTFAID CDSS. In relation to the long-term research environment, signals and images processing may provide *novel representation features*, i.e. non-standard parameters which can add more insight in heart failure domain, for example giving new methods for early diagnosis and prognostic stratification. Actually advanced signals and images processing may be used to turn some medical problems into pattern recognition problems, that can be eventually be solved by KDD or other computational intelligence procedures.
However, raw data or persistent data (which are the starting point for data processing) may not be available depending on the capabilities of the medical devices used in the HEARTFAID scenarios or, even, the network infrastructure of the medical premises. Of course, the ideal situation will be a situation in which each device is able to transfer the acquired raw data together with all the measurements and processing results executed by the software and operators working on the device itself, but unfortunately the scarce adoption of standards (communication protocol and data format) by medical device manufacturers (especially in the non-imaging domain), makes the real-world situation significantly different from the ideal one.
The HEARTFAID philosophy however has been to make a "best effort" integration for each device used in the validation sites in relation to the HEARTFAID project. Thus, for each device all data available in an intelligible format will be stored in the HFP and will be the foundation for the Decision Support System.

Thus, the acquired data made available by the middleware and referred to examinations/vital signs acquired by medical devices will be in different forms:

i. Raw imaging data that can be or not accompanied by some additional data (measurements or features) automatically extracted by the software running on the medical device itself and/or edited by the operator working on the medical device itself. The medical device usually transmits (using a network interface) the examination results (in forms of raw data plus additional information) to the HFP using the DICOM standard. Different ways of interfacing an imaging medical device with the HFP have not been considered due to the fact that they are not standard and they follow a proprietary implementation that requires a huge effort in terms of implementation and the full availability of the necessary information from the manufacturer. Furthermore the large diffusion of the DICOM standard guarantees that most of modern imaging device present in the market can be integrated in the HFP.
ii. Raw non-imaging data that can be accompanied by some additional data (measurements or features) automatically extracted by the software running on the medical device itself and/or edited by the operator working on the medical device itself. The examination results are transferred from the medical device to the HFP in different ways depending on the capabilities of the medical device itself (in case of a medical device that has the basic capability of transferring raw data to a host system in an interoperable way).
iii. Data acquired from a non-imaging device unable to transfer the raw data in a known or standard format. In this case only a final report or some measurements/features extracted by the examination will be available to the HFP.
iv. Data inserted (using specific web form) from the clinicians that are specific to examinations that cannot be transferred in any intelligible format to the HFP from the medical device itself.

1.3 Decision Support Services in the HEARTFAID Platform

As introduced in the previous sections, the HEARTFAID CDSS will be devised to support the medical personnel in their daily activity, by exploiting patients' data and formalized medical knowledge.
The general functionalities that can be assigned to this system are:

- Monitoring system
  - Alerts/reminder/recommendations generated as one-way messages
- Interactive consultation system
  - Engage user in a dialogue
  - Provide recommendations
  - Allow exploration
- Clinical Support System
  - Offers commentary on proposed/actual interventions.

The final aim consists of improving effectiveness and efficiency of the decisional processes involved in the overall clinical management of HF patients, i.e. diagnosis, prognosis, therapy planning and follow-up.
Due to the complexity of this task, the driving force for its accomplishment will be the high potentiality of problem representation and solution that characterize the models and the methodologies of the *Decision Science*. As a result, different methodologies (e.g. computational modelling, knowledge discovery methodologies, visualization and imaging techniques, and the medical knowledge of the relevant domain), will be combined to develop pattern recognition in historical patients' data, signal and images processing methods, computational and inference reasoning.
As a result, HEARTFAID CDSS will be characterized by an innovative and notable feature: it will support clinical decision making by hybridizing (*symbolic*) knowledge based inference with advanced computational (*sub-symbolic*) models.

The activity required for designing and developing the HEARTFAID CDSS will consist in an incremental and evolving process, which can be sketched as shown in Fig. 3. The main stages to be followed are:

- **Identification**: aimed at analyzing the characteristics of the problem at hand, identifying the overall objectives (*requirements*) of the HEARTFAID CDSS, i.e. what it should accomplish and why, specifying the element of the knowledge to be formalized;
- **Conceptualization**: focused on the identification of the general capabilities (*functionalities*) that the CDSS should have to respond to the requirements; characterization and elicitation of the key concepts and relations among the knowledge elements specified at the first stage; identification of the independent modules that must be developed to face the problem;
- **Formalization**: devoted to the selection of the languages and tools for developing the CDSS functionalities. In particular, the language for knowledge representation and the structures used to organize the knowledge;
- **Implementation**: dealing with the development of the HEARTFAID CDSS components and, in particular, with the formalization of the domain knowledge, the definition of the computational models, and the design of the control strategy to work with them.
- **Test**: concerning the evaluation of the system performance, and the critic analysis of the users**.**

However, it should be notices that usually the evolution of the system is not a rigorously linear process, but rather a cycle consisting of continuous refinements (the dashed lines in Figure 1.3): there should be a continuous process of system revision, concepts and models reformulation, and knowledge reorganization which would lead to its successive refinements. Moreover, before accomplishing the formalization stage, relevant issues should be examined, namely the availability of suitable tools for the developing HEARTFAID CDSS; the

technological facilities already available and used by the decision makers, i.e. the clinicians, and the convenience of ad-hoc versus integrated approaches.

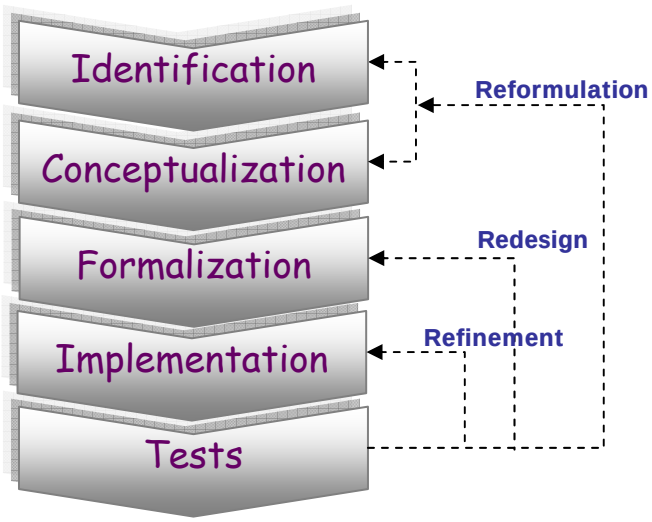


Figure 1.3. *The HEARTFAID CDSS building process*

The main focus of this document is represented by the *Identification* and *Conceptualization* tasks. According to what discussed above, their accomplishment requires, first of all, a deep investigation of general-purpose DSS and of the fundamentals aspects inherent to CDSS, in order to extract all the useful methodologies relevant for facing the HF decisional problems. A survey of such investigation is reported in the third chapter.

2. Data Processing in the HeartFaid

We start this chapter discussing the diagnostic resources selected by the medical partners distinguishing between signal acquisition modalities and imaging modalities. General data processing tasks are then introduced, emphasizing the specificities typical of each of the considered modalities. We mention finally the open problems in the determination of these clinical parameters.

2.1 Diagnostic Resources

2.1.1 Signal Acquisition Modalities

In the field of non-imaging devices or "signal acquisition modalities" from the studies performed in WP1 and WP2 the following modalities have been considered for their use in the HEARTFAID platform:

1) Electrocardiogram (ECG)
2) Holter electrocardiography (with heart rate variability (HRV))
3) Exercise testing:
    a. Exercise test
    b. Cardiopulmonary stress testing
    c. 6-minute walking test
4) Spirometry
5) Bioimpedance
6) Autonomic parameters other than HRV:
    a. Blood pressure variability
    b. Baroreflex sensitivity
    c. Cardiac autonomic modulation parameters
7) Additional devices for specific measurements (only for research workflows)

Introduction to electrocardiography

The heart is organized in atria and ventricles, left and right parts, valves and electric conduction paths.

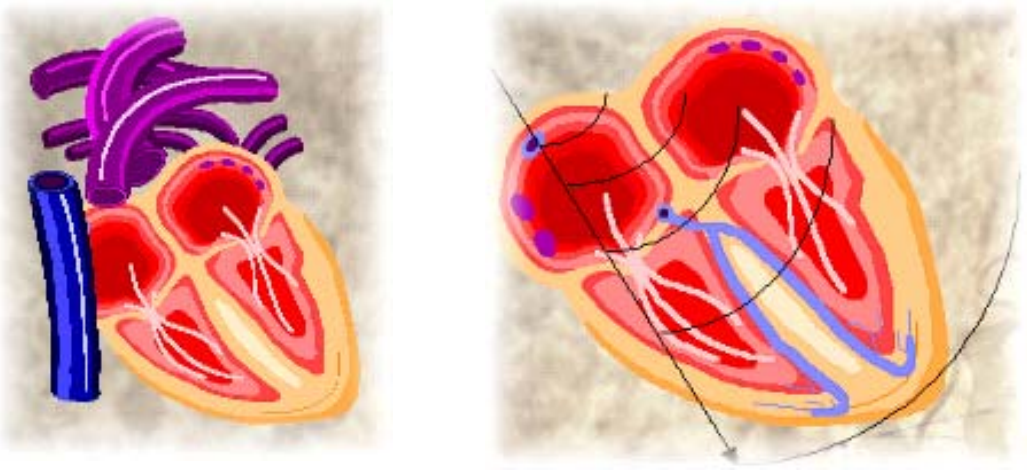


Figure 2.1. *The heart is a mechanical pump controlled by electrical paths*

It is a mechanical pump: it pumps the blood in the systole and relaxes in the diastole. This mechanical pump is controlled through electrical paths.

The electricity starts in the sinus node, it is conducted through the atria to the atrio-ventricular node where the pulse is delayed and after the His bundle the electrical path is divided in two branches (left and right) by the Purkinje system. Once the pulse is started, it is propagated to all the adjacent cells, so the action potentials rise and decrease creating a rather simultaneous electrical phenomenon.

Myocardial activation is accompanied by different potentials in different places of the heart that generate currents that spread all over the body. These currents are modulated by the surfaces interposed.

Generically speaking, electrocardiography is the art of analyzing this electrical activity by measuring potentials at the surface of the body resulting from this electrical activity within the heart (Macfarlane and Lawrie, 1989).

A typical electrocardiogram (ECG) is shown in Figure 2.2.

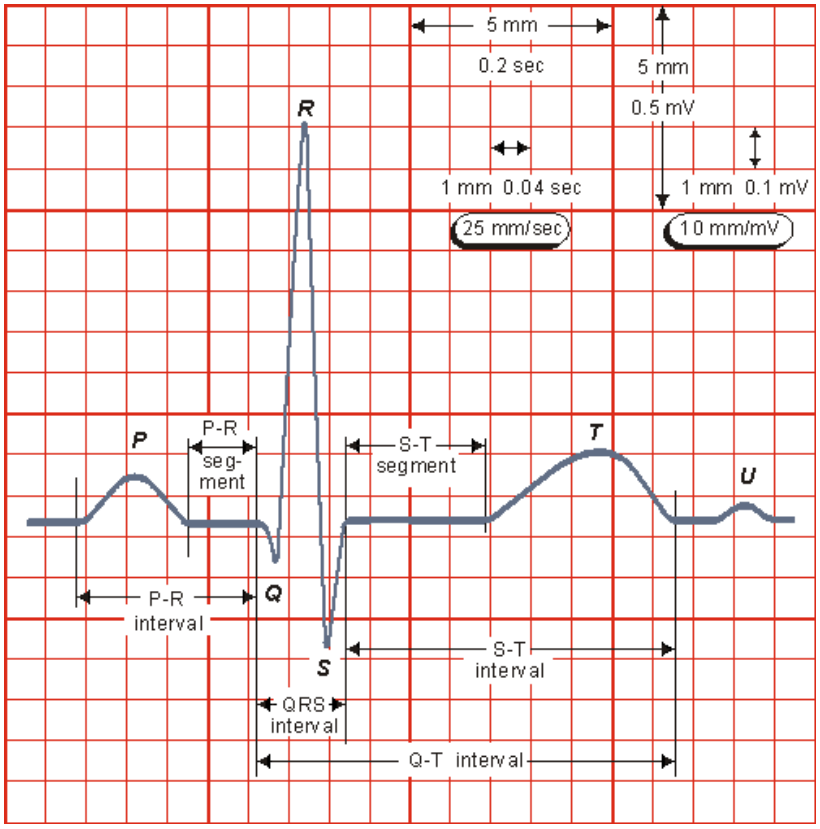


Figure 2.2. *A typical normal ECG waveform*

The P wave corresponds to the atrial depolarization (contraction), then before the Q wave, atrial repolarization may occur. The QRS complex corresponds to the ventricular depolarization (contraction), while the ventricular repolarization corresponds to the T wave.

The electric field generated in the heart can be projected in three planes: frontal plane, sagittal plane and transverse plane.

The non-invasive electrocardiographic signal is acquired through the positioning of electrodes on the surface of the human body. The electrode positioning and the system of electrode placement (electrode configuration) have been standardized during the last century and, based on them, different ECG leads (standardized as well) can be acquired (Malmivuo, 2007a-b).

Electrocardiogram (ECG)

The most common electrocardiographic modality is the so-called 12-channel resting ECG (Willems *et al.*, 1990).

The 12 channels correspond to different points of view of the heart:

- Limb leads (frontal plane). They correspond to a global vertical view of the heart. They can be divided into:
  - Bipolar leads (I, II, III) (Einthoven)
  - Unipolar leads (aVR, aVL, aVF) (Goldsberger)
- Precordial leads (transverse plane). They correspond to a global horizontal view of the heart:
  - Unipolar leads (V1, V2, V3, V4, V5, V6) (Wilson)

According to the European Society of Cardiology some main remarks that can be evaluated from a resting ECG are:

- *"Negative predictive value of normal ECG to exclude left ventricular systolic dysfunction exceeds 90%"*
- *"Anterior Q-waves and LBBB in patients with ischemic heart disease are good predictors of reduced LVEF"*
- *"ECG is very insensitive, albeit specific, for the diagnosis of left ventricular hypertrophy (Sokolow-Lyon)"*
- *"Detection of atrial fibrillation or flutter (causative or contributing factors to HF)"*

From the resting ECG several features are usually extracted by the clinicians to be used in the Heart Failure evaluation (Macfarlane and Lawrie, 1989).

Nowadays several ECG device manufacturers produce "interpretive resting ECG devices" that are able to automatically calculate the main measurements on the ECG signal and a diagnostic proposal.
The technology in a modern ECG device allows:

- Digital acquisition of the ECG
- Digital preprocessing and filtering
- Real-time display and/or printing of the acquired ECG
- Interpretation of the acquired ECG (or quantitative electrocardiography)
- Storage of the ECG and its related information
- Transfer of the ECG to a computer (by direct serial line, modem, floppy disk, memory card, network or wireless, depending on the brand and model of the ECG device)

Thus, commonly all signal processing is performed in the medical devices itself and then the ECG signal, with the result of the processing (measurements and diagnostic proposal) can be transferred to a host computer either to a generic third-party software that implements the implements the communication protocol and knows the data format used by the medical device software for the storage of the examinations or to a software produced by the medical device manufacturer itself.

In all this procedure there is a major problem. Even if standards (i.e. SCP-ECG (EN 1064:2005) or HL7 aECG (Brown *et al.*, 2007)) are made available from the international Standardization Development Organizations (SDO), several manufacturers do not implement or do not properly implement these standards, thus practically hampering the possibility of developing software able to interface their medical devices and to properly read the acquired examinations (see Reynolds *et al.*, 2004).

Holter electrocardiography (AECG) (with heart rate variability (HRV))

The most common Holter ECG modality is the so-called 3-channel 24-hour Holter ECG. It is also called Ambulatory Electrocardiography (AECG) (Morganroth, 1985).

Differently from the resting ECG this modality is a "long-term" ECG recording where a portable ECG recorder (data logger) with some electrodes is installed by the medical personnel on the patient and removed the day after. A Holter monitor is a machine that continuously records the heart's rhythm. Commonly the ECG is recorded during the patient's usual daily life and patient goes home after the installation of the ECG data logger. Electrodes (small conducting patches) are stuck onto the patient's chest and attached to a small recording monitor. The patient carries the Holter monitor in a pocket or in a small pouch worn around your neck or waist. The monitor is battery operated. Usually the patient is requested to accurately record his symptoms and activities so that the doctor can match them with the Holter monitor findings.

Long-term ECG recordings are the basic input for HRV analysis that cannot be performed in recordings shorter than 5 minutes. HRV variability according to the ESC has the following properties in relation to the HF domain:

- *"A marker of autonomic balance, a balance that is characterized by an increased sympathetic activation and reduced vagal stimulation in patients with HF"*

Holter monitoring is used to determine how the heart responds to normal activity. The monitor may also be used:

- When starting a new heart medicine
- After a heart attack
- To diagnose heart rhythm problems

It may be used to diagnose:

- Atrial fibrillation/flutter
- Multifocal atrial tachycardia
- Paroxysmal supraventricular tachycardia
- Palpitations
- Reasons for fainting

Normal variations in heart rate occur with activities. A normal result is no significant changes in heart rhythms or pattern. Abnormal results may include various arrhythmias. Changes in the normal pattern of waves formed by the heart's electrical signals may indicate some abnormalities.

According to the European Society of Cardiology some main remarks that can be evaluated from a Holter ECG are:

- *"Holter monitoring may detect and quantify the nature, frequency and duration of atrial and ventricular arrhythmias which could be causing or exacerbating symptoms of HF"*

Commonly all signal processing is performed in the reviewing station itself and then the ECG signal, with the result (or a subset of it) of the processing (measurements, features and results of the HRV analysis) can usually be exported on an external file for its transfer to a third-party system (Zareba *et al.*, 1998; Kemp *et al.*, 1992).
In case raw data of a Holter ECG will be acquired with some additional information so as to reconstruct the signal, signal processing may be useful in HRV analysis, saving a significant amount of time by the physician and helping discovering new potential features in HRV as pointed out by ESC:

- *"The value of HRV in clinical practice still remains to be determined"*

Exercise testing
According to the European Society of Cardiology there are some remarks about the exercise testing:

- *"Exercise capacity with on line gas exchanges measurements has proved to be an important component of risk profile in HF"*

The typical exercise testing used in the management of the HF patient are:

- Exercise test
- Cardiopulmonary stress testing
- 6-minute walking test

Exercise Test (STX)
An exercise test (STX test) is a special treadmill or cycle ergometer test. This test takes an ECG and measures how the patient's heart works during exercise. It is required to inspect if a patient has any cardiac problems during exercise.
A cardiologist and a nurse are usually present during the exercise test.
For the test, ECG electrodes will be attached to the patient's chest for an electrocardiogram of his heart. His blood pressure will be taken during the test.
The patient will start out by walking slowly on the treadmill or cycling slowly on the ergometer cycle. He will be asked to gradually increase his speed and incline (treadmill) or his power (ergometer cycle) according to a specified protocol.

He can stop the test at any time. If at any point during the test the patient feels as if he has had enough, he will give the “thumbs down” signal to the cardiologist. In case of treadmill, they will slow down it and stop the test. In case of ergometer cycle, the patient will simply stop cycling.

Exercise ECG testing can be used in the following circumstances:

- Assessing a clinical diagnosis of angina
- Risk stratification after myocardial infarction
- Risk stratification in patients with hypertrophic cardiomyopathy
- Evaluation of revascularisation procedures or drug treatment
- Evaluation of exercise tolerance and cardiac function
- Assessment of cardiopulmonary function in patients with dilated cardiomyopathy or heart failure
- Assessment of treatment for arrhythmia
- Assessment of asymptomatic people in high risk occupations like airline pilots.

Cardiopulmonary Stress Testing (CPX)

A cardiopulmonary stress test (CPX test) is a special treadmill or cycle ergometer test. This test takes an ECG and measures how the patient’s lungs work during exercise. It is required to inspect if a patient has any cardiac or respiratory problems during exercise and to determine how well the heart pumps blood to the muscles.

This information is needed to determine the best exercise plan for an HF patient.

An exercise physiologist, a doctor and a nurse are usually present during the cardiopulmonary stress test.

For the test, ECG electrodes will be attached to the patient’s chest for an electrocardiogram of his heart. The patient will breathe into a breath analyzer that measures the gas exchange in his lungs. An oximeter will be placed on his finger to determine how much oxygen is in his blood while he exercises. His blood pressure and pulse will be taken during the test.

The patient will start out by walking slowly on the treadmill or cycling slowly on the ergometer cycle. He will be asked to gradually increase his speed and incline (treadmill) or his power (ergometer cycle) according to a specified protocol. He can stop the test at any time. If at any point during the test the patient feels as if he has had enough, he will give the “thumbs down” signal to the exercise physiologist and doctor. In case of treadmill, they will slow down it and stop the test.

In case of ergometer cycle, the patient will simply stop cycling.

A typical cardiopulmonary stress test system is shown in Figure 2.3.

Figure 2.3. *A cardiopulmonary stress testing system*

The main prognostic markers obtained from a cardiopulmonary exercise testing are:

- Exercise duration
- Peak VO2 (peak of oxygen uptake)
- VE/VCO2 (slope of increase of pulmonary ventilation relative to carbon dioxide production or ventilatory equivalent ratio for carbon dioxide)

Other prognostic markers are:

- Workload
- Blood pressure
- Heart rate
- Peak VCO2
- Anaerobic threshold
- O2 pulse
- SO2 (saturation of oxygen)
- VE/VO2 (ventilatory equivalent ratio for oxygen)
- Respiratory quotient

6-Minute Walking Test (6MWT)

Walking tests have been around since the 1960s, when the 12-min walk was popularized by aerobics fitness enthusiast, KH Cooper, as a quick and easy fitness test (Enright, 2003; ATS, 2002). Patient's functional capacity can be assessed also through a questionnaire or self-report of how much work the patient can do. But patients differ in their ability to properly recall the required information or may underestimate or overestimate their true functional capacity, so objective measurements, when possible, are usually better than self-reports. Thus, the 6-minute walking test (6MWT) is a modality for the objective evaluation of functional exercise capacity.

The 6MWT is a practical simple test that requires a 100-ft hallway but no exercise equipment or advanced training for technicians. Walking is an activity performed daily by all but the most severely impaired patients. This test measures the distance that a patient can quickly walk on a flat, hard surface in a period of 6 minutes (the 6MWD). It evaluates the global and integrated responses of all the systems involved during exercise, including the pulmonary and cardiovascular systems, systemic circulation, peripheral circulation, blood, neuromuscular units, and muscle metabolism. It does not provide specific information on the function of each of the different organs and systems involved in exercise or the mechanism of exercise limitation, as is possible with maximal cardiopulmonary exercise testing. The self-paced 6MWT assesses the submaximal level of functional capacity. Most patients do not achieve maximal exercise capacity during the 6MWT; instead, they choose their own intensity of exercise and are allowed to stop and rest during the test. However, because most activities of daily living are performed at submaximal levels of exertion, the 6MWD may better reflect the functional exercise level for daily physical activities.

The medical devices used in this test are:

- Pulse oximeter (HR and SpO2)
- Blood pressure meter (i.e sphygmomanometer) (SBP and DBP)

These medical devices are required for the measurement of heart rate, blood pressure and oxygen saturation before (baseline) and the end of the test. The total walking distance has to be measured as well.

The test should be performed indoors, along a long, flat, straight, enclosed corridor with a hard surface that is seldom travelled. If the weather is comfortable, the test may be performed outdoors. The walking course must be 30 m in length. A 100-ft hallway is, therefore, required. The length of the corridor should be marked every 3 m. The turnaround points should be marked with a cone (such as an orange traffic cone).

According to the European Society of Cardiology there are some remarks about the 6-minute walking test:

- *"The 6 min walking test may provide useful prognostic information when walking distance is < 300m. However, for use in clinical setting, the value of 6 min test is unclear".*

Spirometry (SPT)

Spirometry (meaning the measuring of breath) is the most common of the pulmonary function tests, measuring lung function, specifically the amount (volume) and/or speed (flow) of air that can be inhaled and exhaled (ATS, 1995). Generally, the patient is asked to take the deepest breath he can, and then exhale into the sensor as hard as possible, for as long as possible.

It is sometimes directly followed by a rapid inhalation (inspiration), in particular when assessing possible upper airway obstruction. Sometimes, the test will be preceded by a period of quiet breathing in and out from the sensor (tidal volume), or the rapid breath in (forced inspiratory part) will come before the forced exhalation. During the test, soft nose clips may be used to prevent air escaping through the nose.
Some handheld and PC-based spirometers are shown in Figure 2.4.

Figure 2.4. *Handheld and PC-based spirometers*

Spirometers are usually store&forward devices and the examination is acquired and stored in the medical device internal memory (or PC database in case of PC-based spirometers). Then, the acquired medical devices can be transferred (or exported) to a third party system, but communication protocol and data format are usually proprietary and not disclosed.
In case raw data are available they contain all the information for displaying the flow-volume, volume-time and flow-time diagrams.

Bioimpedance (BIA)
The body impedance analyzer is a medical device that allows for a non-invasive analysis of the body composition. With the use of some surface electrodes, the device measures the bioelectric values of resistance (correlated to the body liquids) and reactance (correlated to the body solids) present in the human body.
In the HF domain the more significant parameter extracted by BIA is the total body water.
Typically the medical device is a stand-alone device with the capability of printing a final report or is composed by a sensor that is connected to a PC where some specific software for elaborating the acquired data and evaluating the measurements resides.

Autonomic parameters other than HRV
Various invasive and non-invasive techniques have been evaluated to study autonomic function (Tjeerdsma, 2001; NYMC, 2006). The analysis of HRV is an established non-invasive tool to assess autonomic function. Reduced HRV reflects the summation of the autonomic imbalances typically observed during CHF: excess sympathetic tone, parasympathetic withdrawal, and reduced baroreceptor sensitivity.

Autonomic function testing, based on cardiovascular reflexes, is another method to assess autonomic status; the use of several well defined 'stress' tests (the so called Ewing battery) provide information regarding sympathetic and parasympathetic function. Baroreflex function can be assessed invasively using drugs (phenylephrine [vasopressor] or nitroprusside [vasodepressor]), or non-invasively using a technique which directly measures muscle sympathetic nerve traffic (i.e. microneurography) or a more recently introduced indirect technique which studies the relation between changes in arterial blood pressure and heart rate (i.e. baroreflex sensitivity).
The autonomic parameters other than HRV that have been considered for their use in HEARTFAID are:

- Blood pressure variability (BPV)
- Baroreceptor reflex sensitivity (BRS)
- Cardiac autonomic modulation parameters (CAM)

Blood pressure variability (BPV)
Blood pressure (BP) is essentially variable; it fluctuates continuously because of the influence of spontaneous rhythmical variations associated with the functioning of body systems (e.g. respiration, sleeping-waking cycle, seasonal variations) and the effects of superimposed physical and mental activity and of a large number of other behavioral and environmental factors (eg, posture, exercise, mood, ingestion of food and drink, smoking, talking) (Garcia-Vera *et al.*, 2004). Thus, BP fluctuations may be of very short duration, from seconds to minutes, or of longer duration, and these variations differ from a subject to another.
While short-term pressure variability is usually obtained by intra-arterial (invasively) measurements with high sampling rate, long-term BP variability is usually obtained in a long observation period non-invasively. Thus, long-term BP variability is now frequently evaluated by ambulatory monitoring or self-monitoring techniques and expressed, for example, as the standard deviation of BP readings taken every 10-30 min during 24-hour or daytime ambulatory monitoring or as the standard deviation of BP readings taken every 8-12 hours during 2-4 week self-monitoring (Novakova *et al.*, 2000).
Research on BP variability has evolved following 2 lines. The first line examined the hypothesis that individuals with increased BP variability would suffer more vascular damage. The second research line examined BP variability in the context of the reactivity hypothesis. This hypothesis states that individuals who show increased cardiovascular reactivity to psychologically stressful stimuli are at increased risk of developing hypertension.
In the context of HEARTFAID the blood pressure variability will be monitored through non-invasive Holter BP devices that can be used by patient in their normal daily life (homes, workplaces, etc.). The blood pressure will be typically acquired during a 24-hour period. The blood pressure value will be either transferred to a manufacturer's software running on a PC and then exported typically in a textual format (including the raw data) or directly transmitted to a third-party software (in this case the communication protocol and the data format have to be available).

Baroreflex sensitivity (BRS)

Baroreceptor control of the circulation has generally been thought to be associated with short-term buffering of changes in arterial pressure rather than with long-term setting of pressure levels. Recently, evidence has been accumulated which suggests that impairment of the baroreflex may have an important effect on long-term arterial pressure. Further interest in the baroreflex has been stimulated when several studies have demonstrated that autonomic control, measured by either baroreflex sensitivity or heart rate variability, is a powerful independent prognostic factor which adds to other clinical measures of risk stratification in patients with left ventricular damage or in those who have survived myocardial infarction. Inter-relationships of these mechanisms are complex, poorly understood and subject to many confounding factors (Novakova *et al.*, 2004; Krticka *et al.*, 2000).

The method of baroreflex sensitivity determination usually used is based on spectral analysis of the spontaneous variability of blood pressure and pulse intervals.

In HEARTFAID validation sites, the device usually used in clinical settings for this modality is the "Finometer PRO or MIDI" (non-invasive stationary blood measurements and beat-to-beat haemodynamic monitoring systems) (produced by Finapres). For the normal daily-life monitoring the Portapres device can be used. It is a data logger that can transfer all the data to a PC where the same manufacturer's software used for the PRO or MIDI devices is installed. The manufacturer software that allows the evaluation of all the significant parameters listed above.

In Figure 2.5 Finapres devices are shown.

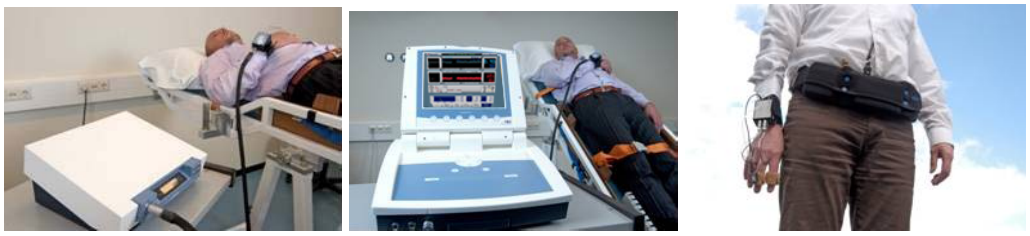


Figure 2.5. *Finapres devices for the measurements of the baroreflex sensitivity*

It seems that the Finapres software is not able to export the raw data and the examination results in a known or disclosed format, thus probably the numeric value of interest will be filled in by the clinicians using a specific web form.

Cardiac autonomic modulation parameters (CAM)

There are other cardiac autonomic modulation parameters that will be collected for research purposes. These values are typically collected with a Holter BP device with the capability of measuring also the HR. The averaged value in the 24-hour period will be stored as indexes of the cardiac autonomic modulation.

Additional devices for specific measurements (only for research workflows)
Additional devices for specific measurements have not been considered at this stage. Only the above list of devices will be used in first instance. Of course, the results obtained from the research workflow might suggest the use of other modalities that have not been considered in this initial design. In that case the integration of any new modality will be made on a best effort base and, if raw data will be available in the HEARTFAID platform, ad hoc signal processing will be designed in order to automatically extract the required features (or part of them) from the raw data (if not transmitted by the medical device together with the raw data) or in order to facilitate the medical personnel in the proper extraction of the required features (or part of them) from the raw data.

2.1.2 Images Acquisition Modalities

Echocardiography
Echocardiography is recognized as a valuable imaging modality for the qualitative and quantitative assessment of heart health. Actually, when compared with other promising techniques like MRI and ultrafast CT, echocardiography is essentially inexpensive and, in a single routine examination, it provides a bunch of information ranging from heart chambers dimension to semi-quantitative assessment of the valvular functions. Furthermore, being real time, it is suitable for deformation analysis and assessment of dynamic functional features of the organ. Nevertheless, the quality of information content is often reduced by noise, speckle, attenuation in the far field, suboptimal acoustic windows and the like. Even worse, due to small impedance difference between blood and endocardium, the endocardial border may be broken or ill-defined. For these reasons, accurate assessment of cardiac parameters depends tightly from the expertise of the sonographer.

Main Standard Views in Transthoracic Echocardiography (TTE)

Echocardiography is carried out using dedicated specialized ultrasound devices. Ultrasounds of different frequencies (2-4 MHz in the adult) are transmitted from a transducer placed on the anterior chest wall of the patient, lying in left decubitus position. Continuous ECG recording is performed to time cardiac events.

The transducer is placed in a number of different positions on the chest corresponding to suitable acoustic windows. In each window, a set of standardized views of the heart is acquired. There is no automatic guidance in the choice of the view that is manually selected by the sonographer on the basis of the direct feedback on the device screen.

1. Left parasternal window (usually 2^{nd} to 4^{th} intercostal space, left sternal edge)
    a. Long-axis view. In this view, the left ventricle (LV), the Left Ventricle Outflow Tract (LVOT), the Left Atrium (LA) and the Right Ventricle (RV) are visible.

   b. Short-axis views. In the same acoustic window, a number of short axis views can be obtained in various planes approximately orthogonal to the LV long axis. The most common planes are in the aortic valve (AV) plane, in the mitral valve (MV) plane, in the mid-papillary plane and in LV apex plane.
2. Apical windows
   a. 4-chamber view. In this view the two atria and ventricles are visible; the view is important in chamber quantification and for assessing valvular functions.
   b. 5-chamber view. Starting with the 4-chamber view and changing the angulations of the transducer, the so-called 5-chamber view is obtained. Actually, the additional chamber is not a real chamber, but it corresponds to the aortic outflow tract. The view is useful in assessing aortic regurgitation (AR) and aortic stenosis (AS).
   c. 2-chamber view. Rotating the transducer, starting with the 4-chamber view, one obtains the 2-chamber view, featuring different segments of the LA and LV.
3. Subcostal window (under the xiphisternum)
   a. It offers similar views similar to the apical window, but rotated by 90 degrees. It is useful in assessing the size of the inferior vena cava (IVC) and its collapsing pattern (a piece of information needed for indirect non-invasive pulmonary pressure estimation).

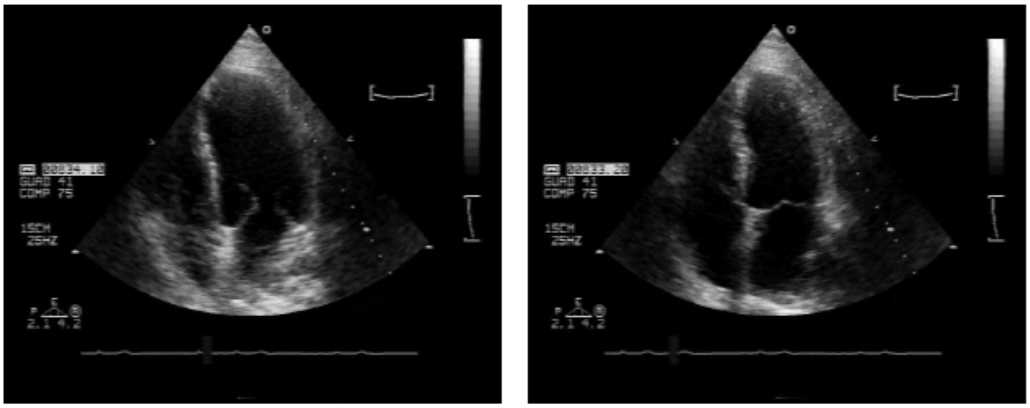


Figure 2.6. *End diastole (a) and end sistole (b) frames from an apical view of the heart*

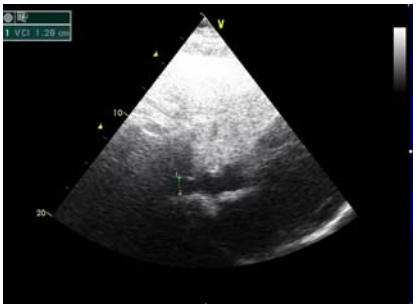


Figure 2.7. *View and quantification of the inferior vena cava (VCI) in the subcostal window*

Related echocardiographic examinations

Other related techniques include **contrast-enhanced ultrasound** or **transesophageal echocardiography (TEE)**. In the first case gas-filled microbubbles are injected into the venous system to improve tissue and blood delineation and improve flow-related measurements. In TEE examination the heart is imaged from a transducer introduced in the esophagus using an endoscope. It conveys clearer images, since lung tissue does not intervene between the heart and the esophagus. Further a shorter field of view is enough in TEE, thus permitting the use of higher frequency transducers. However, TEE is an invasive unpleasant examination. Therefore its use is limited to those patients having inadequate echo windows.

Echo techniques

There are mainly 4 echo methods used in clinical setting: two dimensional echo, M-mode, Doppler, and three dimensional echocardiography.

1. 2D echo gives a snapshot in time of a cross section of tissue; if these sections are produced in quick succession, we have a 2D image sequences, showing in real time the heart chambers, vessels and valves during the deformation cycle of the organ. Some examples of 2D echo images are shown in Figure 2.6 and Figure 2.7.
2. M-mode refers instead to ultrasound data acquired along a single direction and acquired over time. In this case, a 2D image is displayed, having in the x direction time and in the y direction the recorded intensity of the tissue along the chosen direction. M-mode echocardiogram are used mainly for linear measurement (for example for quantification of chambers size, valvular orifice and thickness of heart walls) and for semi-quantitative assessment of regional wall motion. The choice of a suitable beam direction is crucial in most applications; actually the beam should be oriented perpendicularly to the structure to be analyzed. To achieve this, guidance by 2D echo is used (2D targeted M-Mode). An example of M-mode image is shown in Figure 2.8.

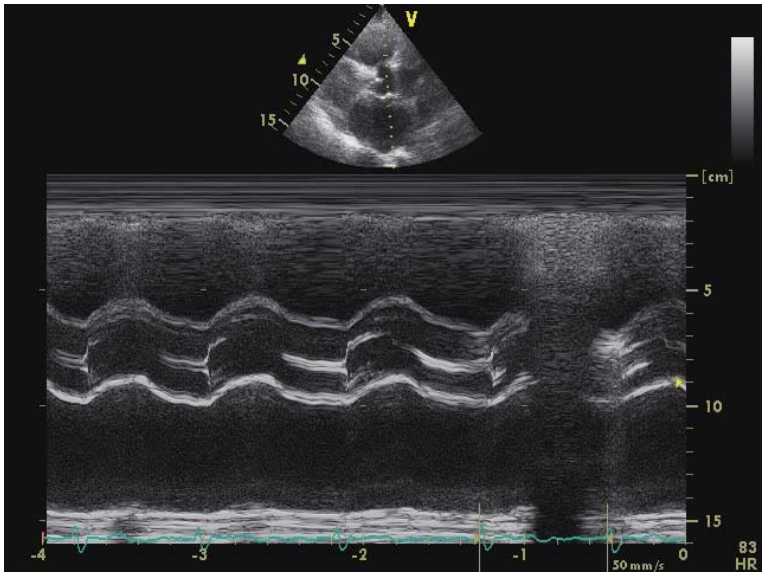


Figure 2.8. *Example of 2D-targeted M-mode*

3. Doppler echo uses the Doppler shift principle to determine the velocity and direction of blood flow or tissue (in Tissue Doppler Imaging (TDI)). Two kinds of receiving/transmitting techniques are normally available on any ultrasound device: continuous wave Doppler and pulsed wave Doppler. Continuous wave Doppler (CWD) uses two ultrasound crystals, one transmitting and the other one receiving. High blood flow velocities can be accurately determined, however the ability to localize precisely the flow signal is limited, since the signal is recorded on the full length of the ultrasound beam. By converse, pulsed wave Doppler (PWD) uses only one crystal to transmit and receive after a suitable time delay. The time delay determines the depth at which the signal is recorded, thus allowing for optimal spatial focus. On modern devices, the sonographer sets manually a small region of interest in a 2D echo image and then the device use an appropriate time delay to acquire PWD data on the selected region. The drawback of PWD is limited capability to record high velocity; indeed for velocity whose associated shift is beyond the Nyquist frequency (corresponding to one half of the inverse of time delay), the phenomenon of aliasing occurs. PWD is used to get colour flow mapping, which is conventional 2D images having superimposed the velocity of tissue or blood flow. The velocity is coded in colours ranging from blue to red according to the BART convention (blue away, red towards). An example is shown in Figure 2.9.

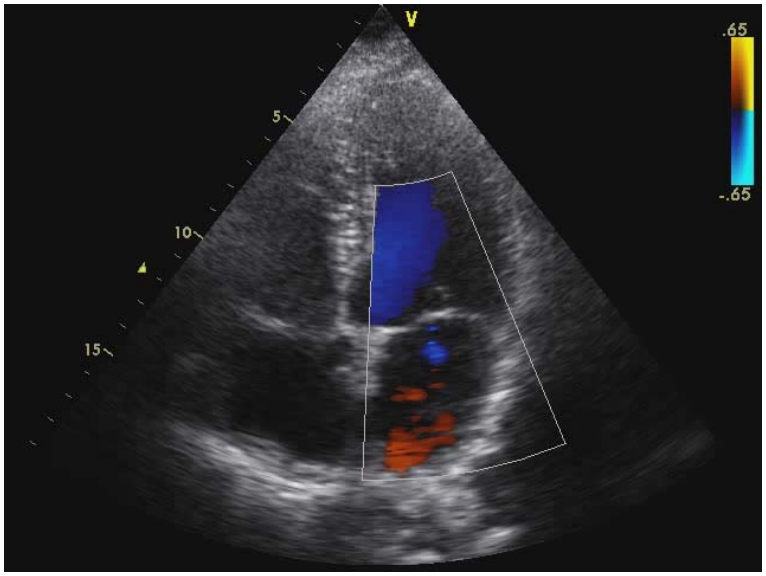


Figure 2.9. *Apical 4C view; Doppler values are shown according to BART convention in a region around the mitral valve*

Three dimensional echocardiography is still a technique under development, used mainly in medical research environment. Early attempts to 3D echocardiographic imaging where based on 2D imaging augmented by a tracking system of the transducer in a single acoustic window. Postprocessing was then used to get a meaningful 3D image by fusing the acquired 2D views (Geiser *et al.*, 1982; Moritz *et al.*, 1983). More recently, native 3D transducers, consisting in a two dimensional array of transducer, where employed (Takuma *et al.*, 1999) to acquire a pyramid image. The main advantage consists in an image acquired in only one cardiac cycle, instead of being assembled from various cycles as in previous attempts. A second generation of native real-time 3D echocardiography systems has been produced by Philips (Franke and Kohl, 2003), providing higher quality data. The 3D images could be used to explore the heart in any cross section, by voxel reconstruction. Visualization of cardiac structures by surface reconstruction is also feasible, but up to now requires tedious manual border tracing in multiples image planes. For these reasons, 3D echocardiography is particularly suitable for clinical research to get accurate quantification of chamber size and to detect valvular defects, but it is still too expensive for routine clinical examinations. We refer to ASE position paper (Hung *et al.*, 2007) for an up-to-date status of 3D echocardiography and a discussion of guidelines for appropriate application of this technique based on available evidence.

Echocardiography in the heart failure domain

According to ESC guidelines (Swedberg *et al.*, 2005), echocardiography is the preferred method for the documentation of cardiac dysfunction at rest; TTE is encouraged for the diagnosis of heart failure, being a non-invasive, rapid and widely available technique.

It allows for the assessment of chamber dimensions, wall thickness, motion and geometry, indices of regional and global, diastolic and systolic ventricular functions. The most important measurement is the left ventricular ejection fraction (EF), that permits to discriminate between patients with preserved or impaired systolic function. In latest years, the assessment of LV diastolic function has been recognized as in important examination in the heart failure domain. In particular, in combination with usual Doppler echo, TDI is currently employed in some HEARTFAID validation sites to get a precise characterization of the LV filling pattern.

Chest X-Ray

Chest X-ray, which dates back to the early of XXth century, is one of the oldest medical imaging modalities.
In chest radiography, photons are projected through the thorax to an X-ray-sensitive detector (film or sensor). Dense structures (such as bones and water) absorb more energy than air-filled tissues. In a normal lung, approximately 2/3 of the total volume is made up of air; blood and water account 1/6 each of this volume. Thus the medium density is 0.3 g/ml.

In the heart failure domain, according to ESC guidelines, the chest radiography should be part of the initial diagnostic work-up.

However, high predictive value of X-ray findings is only achieved by interpretation of X-ray in the context of clinical findings and ECG anomalies. It may be useful to detect cardiomegaly, pleural effusion, pulmonary congestion, interstitial and alveolar edema and other diseases contributing or causing dyspnoea.

- *Cardiomegaly* (i.e. the abnormal enlargement of the heart; see D5 for a physiopathological discussion) is usually assessed considering the so-called *cardiothoracic ratio*, a parameter easily determinable by linear measurement on the chest X-ray image.
- In a normal lung, pleural space is lined by a thin film of fluid. The visceral and parietal pleura are in opposition and glide over each other during the respiration. *Pleural effusion* is the phenomenon consisting in fluid accumulation in the pleural space. Common causes of pleural effusion are hydrostatic pressure changes as in CHF, cirrhosis and hypoalbuminemia. Pleural effusion may be easily detected by chest X-ray. For example, in an upright chest X-ray, pleural effusion is detectable by a high intensity homogeneous region at the base of right lung.

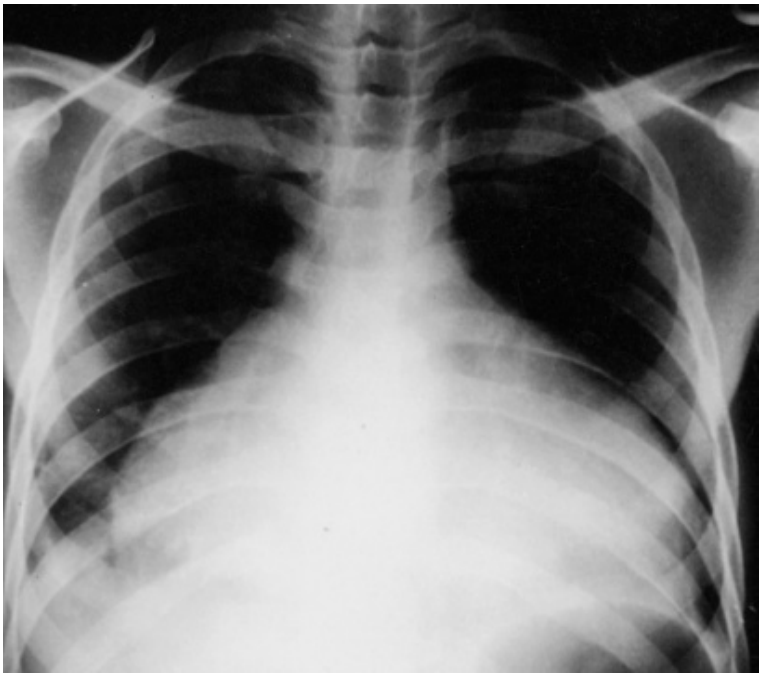


Figure 2.10. *Severe cardiomegaly*

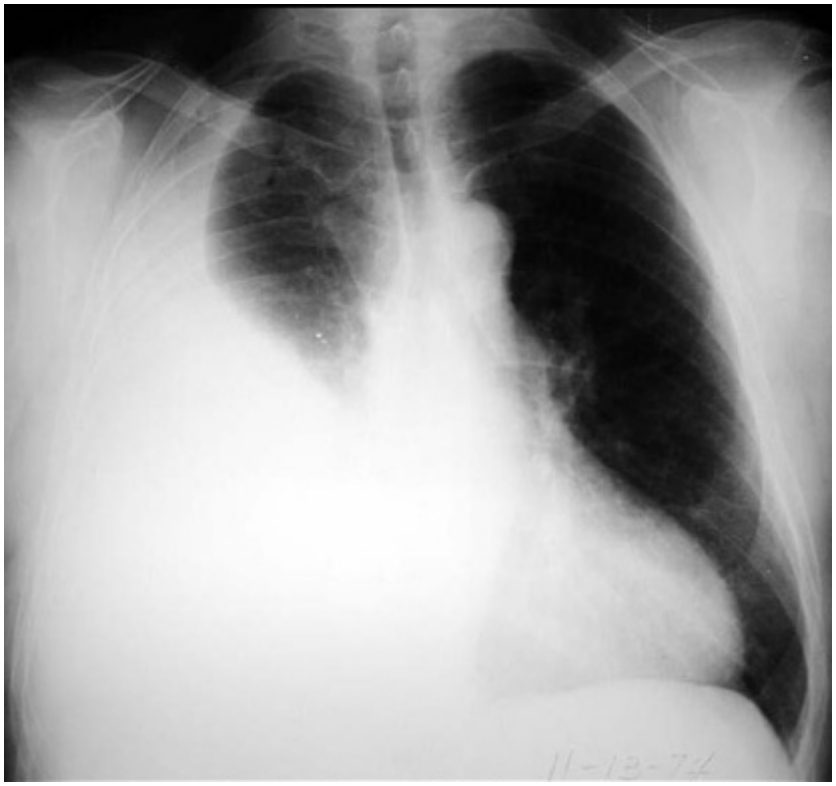


Figure 2.11. *Pleural effusion, as seen in an upright chest X-ray film*

- Pulmonary oedema is the swelling or fluid accumulation in the lungs. Usually it could be suspected by the symptoms reported by the patient or by the findings of a physical examination. The confirmation of the diagnosis is

obtained by conventional chest X-ray. On a radiographic film, small high intensity regions denote fluid accumulation in the alveolar walls.

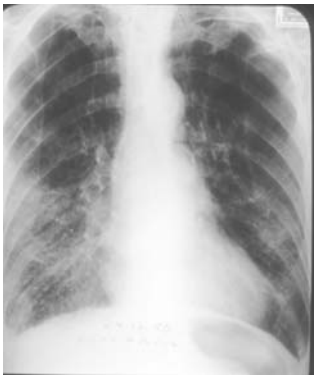


Figure 2.12. *Evidence of pulmonary oedema in a chest X-ray image*

- Connected to interstitial pulmonary oedema, Kerley B-lines are often detectable by chest X-ray. They are represented on the film as linear opacities, being 1-2 cm long, in horizontal direction meeting the pleura at right angles. They are typically seen as a ladder up the side of the lungs beginning at the costophrenic angle. Kerley B lines represent interlobular lymphatics which have been distended by fluid or tissue and they are usually an indication of raised pulmonary venous pressure. As such, they can be a further indication of Congestive Heart Failure or mitral stenosis.

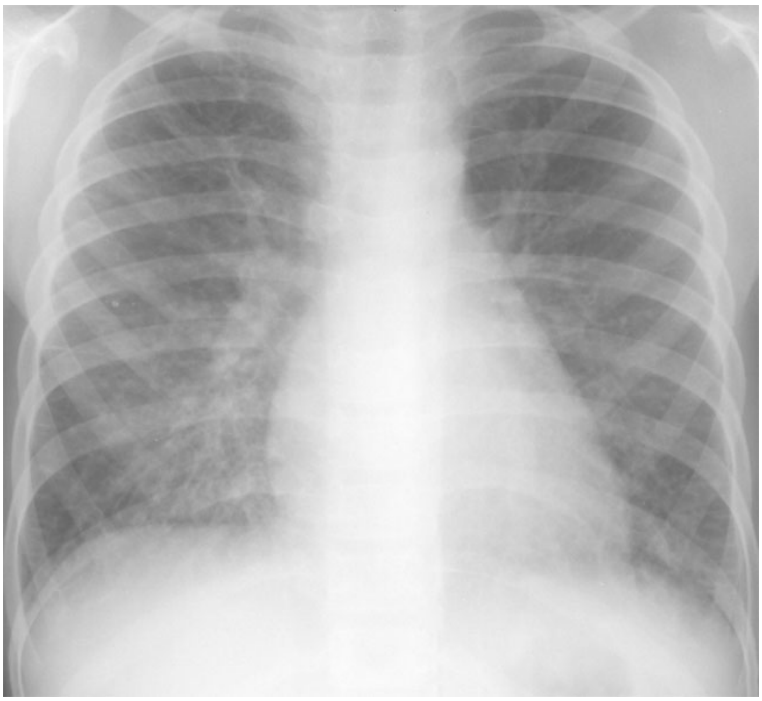


Figure 2.13. *A chest X-ray showing Kerley B lines*

- The mean vascular pedicle width (see Figure 2.14) is highly correlated with total blood volume, which is of interest in some pathology: This measure can be easily performed on conventional radiograms; alternatively, dedicated software for automatic analysis of digital radiograms can be implemented. The precision and the reliability of the measurements performed on chest radiograms depend on the presence of artefacts, caused by the formation mode of radiographic images and depending either on the shape of the X-ray beam or on the patient's position during X-ray exposure. Thus, in order to obtain accurate and repeatable measurements it is mandatory to observe appropriate protocols during radiograms exposure.

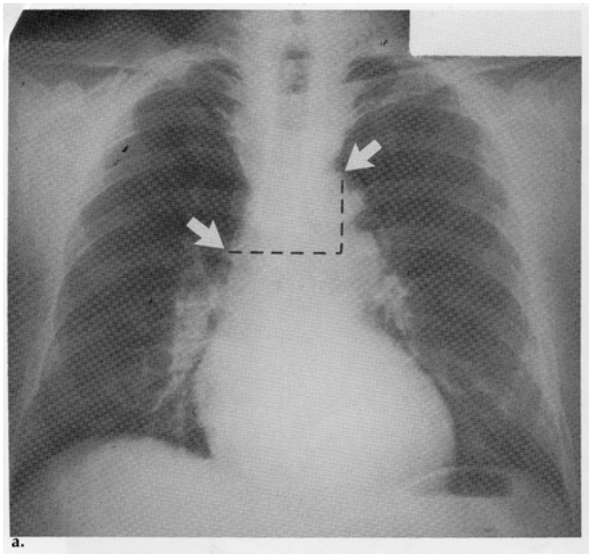


Figure 2.14. *Measuring points for vascular pedicle width. Left: the point at which the superior vena cava crosses the right main bronchus; right: the point of the takeoff of the left subclavian artery from the aorta.*

Magnetic Resonance Imaging (MRI)

According to ESC guidelines, MRI is a versatile well-established imaging technique that has become the *de facto* gold standard in accuracy and reproducibility for the assessment of heart volume, mass, motion, thickening and deformation. Further it allows detecting congenital defects, tumours and valvular and pericardial diseases. It is less operator-dependent when compared to TTE.
However, MRI has not been shown to be superior to TTE in the practical management of HF patients. So, being expensive, it is not routinely performed; its use is limited to patients with inadequate acoustic windows that require specific parameters determination.
Nevertheless it is likely that in the following years MRI will become a less rare resource. The insertion of these imaging modalities in the HEARTFAID platform would be a perspective value added at the platform, since computer vision techniques could be successfully applied to MRI imaging sequences in order to

convey a unified picture of the state of the heart, suitable for precise classification by the HEARTFAID DSS. Furthermore, in recent years, MRI-based imaging modalities have been developer by the University of Pennsylvania and Johns Hopkins (Tagged, SPAMM and HARP imaging)(see Kahlifa *et al.* (2005) and bibliography therein). The basic idea underlying to the works of these research groups is to *decorate* tissues with magnetic patterns. Since magnetization is a property of matter, the observed deformation of the induced magnetic pattern could be directly exploited to analyze motion and deformation of the heart, thus giving more insight for medical research and mechanical cardiac modelling.

Additional imaging modalities

When TTE at rest was incapable to provide enough information or in case of evidence of coronary artery disease, additional non-invasive techniques may be considered. They include: stress echocardiography, ultrafast CT, radionuclide angiography, SPECT, PET. They have not been considered in previous deliverables.

2.2 Principles of Signal and Image processing and Mathematical Models for Features Extraction

In this section, we give a general overview of signals and images processing and we discuss the tasks more relevant for HEARTFAID platform.

2.2.1 Principles of signal processing

Signal processing is the analysis, study, manipulation, interpretation of signals of various nature, ranging from radar images to biomedical signals, which are our present topic.
Processing of such signals includes storage and reconstruction, source separation, noise reduction, compression and feature extraction.
A basic mathematical tool useful to make easier many of the aforementioned signal processing tasks is the *Fourier transform*. After the application of this transform, the analysis shifts from the *time domain* to the *frequency domain*. In fact, the signal is no longer represented as a succession of values as a time function, but as a set of coefficient pairs as a function of frequency, each pair consisting of an *amplitude* and a *phase*. Some methods of analysis are typical, however, of the time domain: simple examples are given by the calculation of parameters describing the *amplitude distribution* of the signal, such as its *statistical moments*.

The mathematical model underlying the Fourier transform is based on the idea that any signal is an element of a vector space (actually a *Hilbert space*) and that a *basis* of this space is given by the set of unitary vectors uniformly rotating with all possible speeds. Each signal can be represented as the sum (or series, or integral) of components along the vectors of the basis.

Let $v_f(t) = exp(2\pi ift)$ be the basis vector rotating with frequency f (t is time and i the imaginary unit). The value of the Fourier transform of a signal $s(t)$ for frequency f is the *scalar product* of the signal and $v_f(t)$, defined as:

$$F(f) = \langle v_f(t), s(t) \rangle = \int v_f^*(t)\, s(t)\, dt\ .$$

Each frequency component $\gamma(f, t)$ is given by the product $[F(f)\ v_f(t)]$, and the original signal can be developed as the integral of its components:

$$s(t) = \int \gamma(f, t)\, df.$$

The Discrete Fourier Transform (DFT) is applied to a discrete signal $s[n]$, which can be obtained by sampling a continuous signal with sampling interval T. This transform is defined as:

$$F_D(f) = \Sigma_n\, s[n] \exp(2\pi infT);$$

it is periodic (obviously in the frequency domain) with period equal to the sampling frequency $f_s = 1/T$.

It should be taken into account, however, that a number of analyses can be carried out in both domains: the choice is usually based on reasons of computational costs. For instance, *filtering* can be performed either by multiplication in the frequency domain or by convolution in the time domain; *correlation* can be calculated either by conjugate multiplication in the frequency domain or by sign change followed by convolution in the time domain. We will consider these two important operations below more in detail.

Energy and *power* are computed by integration in any of the two domains. In fact, Parseval's Theorem states that the energy, defined in the time domain as the integral of the squared signal, can also be computed by integrating the squared amplitude of the Fourier transform in the frequency domain.

There are other important correspondences between descriptors defined in one domain and descriptors defined in the other. For instance, the *number of baseline crossings* per time unit provides information very similar to that given by the *centre of mass* of the Fourier spectrum: in fact, this number (divided by two) can be applied for a time domain estimation of the central frequency of a signal.

A number of methods consist in an efficacious combination of time-domain and frequency-domain stages. For instance, the envelope of a signal can be easily approximated by rectifying the signal in the time domain and then applying a *low-pass filter* in the frequency domain. In this way, the signal is studied as the result of a process of *amplitude modulation*.

In addition to the Fourier transform, other transforms are useful, in particular the Wavelet functions. These transforms allow representing the signal in both domains at the same time: an interesting property, considering that a number of events can be seen as time-transient changes in signal frequency. The most commonly applied discrete wavelets are: Coiflet, Daubechies, and Haar, while the

most common continuous wavelets are: Mexican, Hermitian, Morlet, and Beta. It should be recalled, however, that also the so-called windowed Fourier transform provides information both in time and frequency.

As an example, Figure 2.15 shows the shape of a Morlet wavelet. By changing the horizontal (time) scale of the figure we obtain a set of wavelets which are identical from the point of view of morphology but change in their frequency content. By computing the scalar products of a signal with the various wavelets, the entire spectral range of the signal can be explored. Furthermore, since each Wavelet only lasts for a short epoch, time information is provided in addition to frequency information.

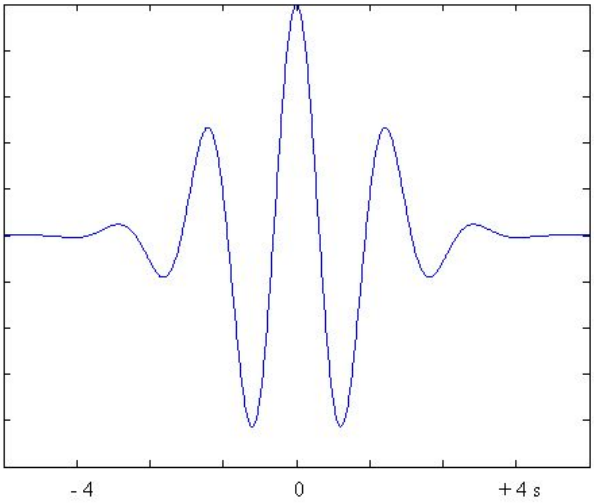


Figure 2.15. *Mortlet Wavelet*

Filtering is a basic tool for signal processing: it is applied for numerous purposes. First, any kind of digital signal analysis requires a preliminary *anti-aliasing* filtering, in order to remove the high-frequency components that would imply a violation of the Nyquist-Shannon theorem. In fact, according to this theorem, frequencies higher than one-half of the sampling frequency make it impossible to reconstruct the original signal. Moreover, signal preprocessing includes the application of a *band-pass filter* able to remove low-frequency noise consisting in baseline wander and high frequency noise (such as muscle activity in the ECG signal). Another application of filtering is selective band removal: typically, power-line interference at 50 or 60 Hz must be filtered out. At last, important waveforms are often characterized in frequency, so that a *narrow-band filter* can facilitate their detection and classification: well-known instances are the Spindles in sleep EEG (usually in the range 14-16 Hz) and the PQRST Complex in ECG: for the latter, the central frequency of the applied filters varies from 5 to 15 Hz. For the implementation of filters, the *cut-off frequencies* are not the only crucial characteristics to choose carefully: *filter slopes* are important, and the properties of the *phase response* as well: *linear-phase filtering* is often to be preferred. Discrete filters are divided into two categories, FIR (Finite Impulse Response) and IIR (Infinite Impulse Response, i.e. recursive filters): each of these

two kinds of filters present advantages and disadvantages, and the choice is generally based on a reasonable trade-off.

A *linear discrete filter* is characterized by its impulse response *h[n]*. The DFT $H_D(f)$ of *h[n]* is called the *frequency response* of the filter, and its squared amplitude is called *gain*. If the input is x[n], and its Discrete Fourier Transform is $X_D(f)$, then the output *y[n]* is given by the *convolution* of *x[n]* and *h[n]*, defined as

$$y[n] = \Sigma_k \, x[k] \, h[n-k].$$

In the frequency domain the output is given by a simple algebraic product:

$$Y_D(f) = X_D(f) \, H_D(f).$$

The equivalence between convolution in one domain and multiplication in the other is stated by the so-called convolution theorem.

Typical linear filters are the Butterworth filters, which are very flat in the pass-band (i.e. they do not present ripples) and roll off towards zero in the stop-band. Low-pass Butterworth filters with large order tend to approximate an *ideal low-pass filter*, i.e. a filter whose shape is given by a rectangle: the gain of an ideal low-pass filter is equal to 1 for frequencies lower that the cut-off frequency and equal to 0 for frequencies higher than the cut-off frequency.

Figure 2.16, Figure 2.17 and Figure 2.18 illustrate the behaviour of an ideal low-pass filter. The signal of Figure 2.16 contains frequencies in the range from 0.01 Hz and 0.06 Hz (the abscissa units are seconds). Applying an ideal low-pass and an ideal high-pass filter, both with cut-off frequency 0.03 Hz, the signals of Figure 2.17 and Figure 2.18 are obtained, respectively. Of course, the sum of these two output signals gives the original signal.

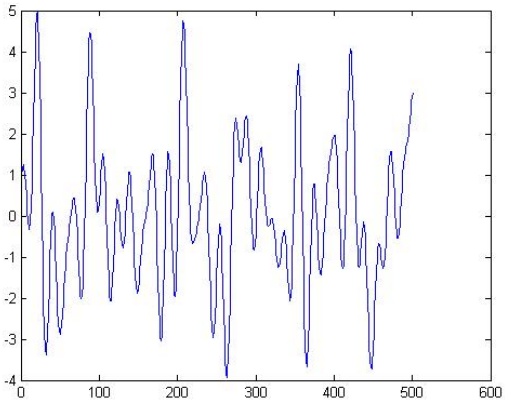


Figure 2.16. *Signal including frequencies from 0.01 to 0.06 Hz.*

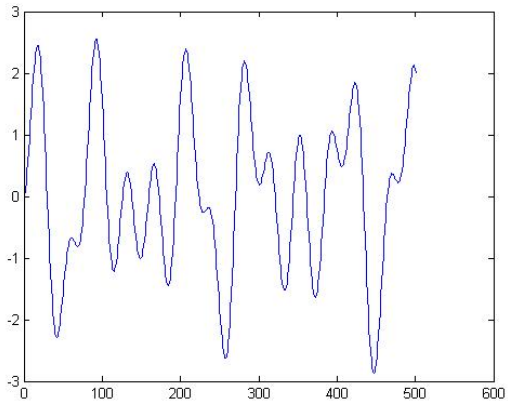


Figure 2.17. *Output of an Ideal Low Pass Filter; the input is the signal in Figure 2.16*

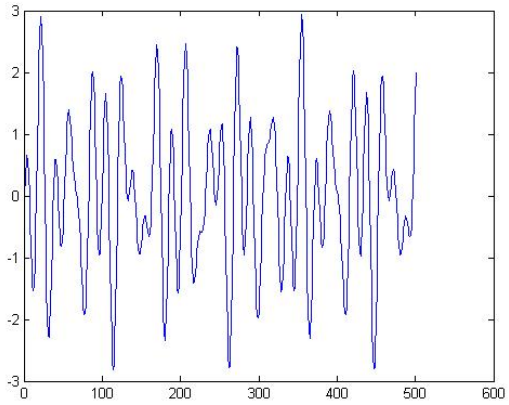


Figure 2.18. *Output of an Ideal High-Pass Filter; the input is the signal of Figure 2.16*

Figure 2.19 shows the impulse response of a discrete filter described by a finite-difference equation (equivalent to a differential equation for continuous signals) of order 2 with damping factor equal to 0.5. This linear filter describes the behaviour of simple mechanical and electrical systems presenting damped oscillations.

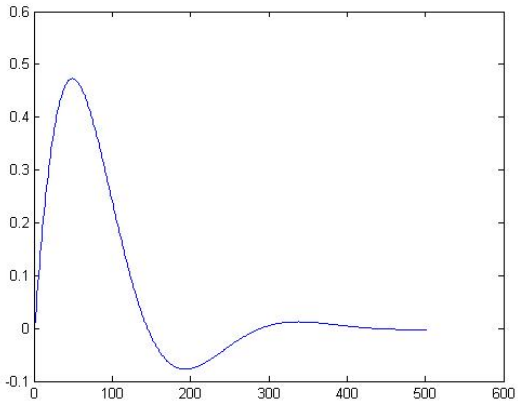


Figure 2.19. *Impulse Response of a linear second-order system with damping factor 0.5*

A classic method for waveform recognition is the application of the so-called *matched filter*: the actual signal is continuously compared with a *template*: this comparison is carried out by the computation of correlation. If the signal is *s[n]* and the template *z[n]*, then the correlation is:

$$c[n] = \Sigma_k \, s[k] \, z[k-n].$$

Clearly, correlation is equal to convolution preceded by a sign change for one of the two signals. When correlation between signal and template is high, the processed signal "resembles" the template, and the waveform is thus identified. Correlation is also important for the comparison of different "channels" of a multi-channel signal, such as the various derivations of the EEG, or the standard 12 channels of the ECG. In the frequency domain, the comparisons between two signals x[n] and y[n] can be performed by the computation of the so-called *cross spectrum*, which is equal to the product $X_D(f)\, Y_D^*(f)$. The absolute value of the cross spectrum, normalized in order to obtain values between 0 and 1, is called *coherence*.

Figure 2.20 shows the correlation of the signal of Figure 2.16 with the same signal with a time shift of 100 s. The correlation presents a maximum exactly for t = 100 s.

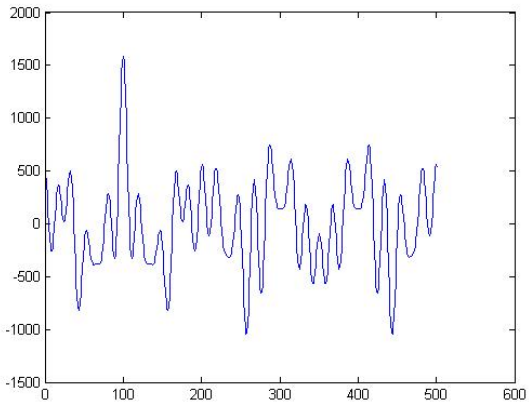


Figure 2.20. *Example of Correlation*

A matched filter can be used in communication systems. For example, as shown in the schema of Figure 2.21 shows, it can be applied in order to recognize the pulses of a binary message which is transmitted across a noisy channel.

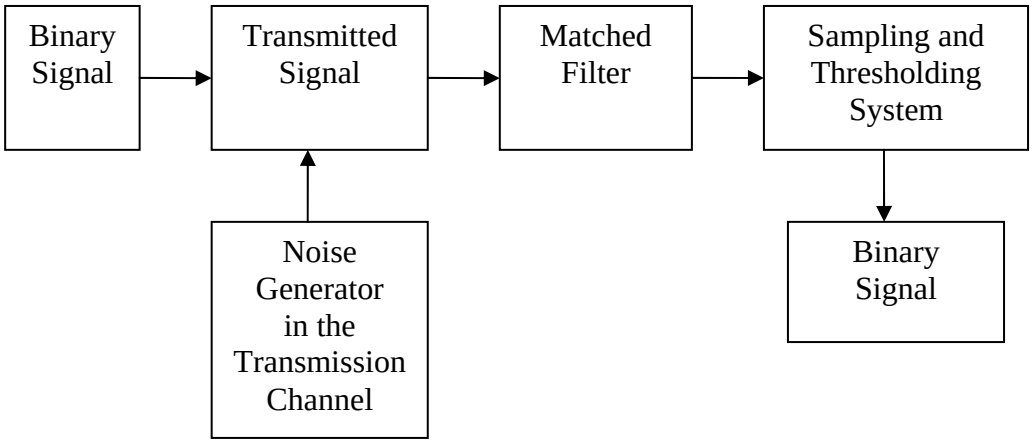


Figure 2.21. *Application of Matched Filters in communication systems*

Considering that correlation is a linear operation, non-linear functions have been introduced in order to identify similarities or dissimilarities between signals that can escape a measure only based on correlation.

There are alternative methods for waveform detection: a simple method is based on the computation of the derivative of the signal and the application of an appropriate threshold: for instance, the QRS end in ECG can by determined as the instant in which the derivative, after assuming the peak value, decreases to a certain threshold value.

Other methods of waveform recognition consist in applying a Maximum-Likelihood (ML) estimator to a statistical model which includes parameters: the actual signal is viewed as the sum of stationary noise and an appropriate waveform characterized by the parameters of the model. ML methods consist in procedures that provide those parameter values that make the known likelihood distribution a maximum: of course, it must be previously assumed that a certain statistics is valid.

Other approaches to the detection of waveforms have been based on *hidden Markov models*. Figure 2.22 shows a general schema for a hidden Markov model. The observed variable *y(t)* depends on the value of a hidden variable *x(t)* at the same time, while the hidden variable enjoys the so-called *Markov property*, i.e. its value at time *t* depends on the value at time *(t-1)*.

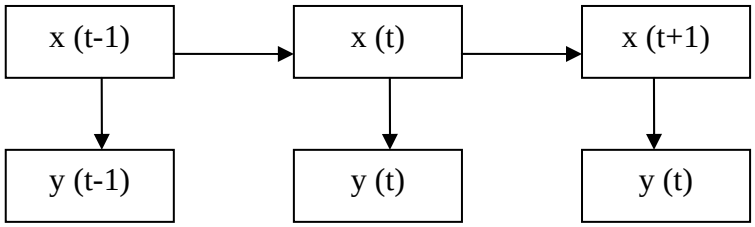


Figure 2.22. *Basic Schema of a Hidden Markov Model*

Furthermore, Artificial Neural Networks (ANN) have been applied as *adaptive non-linear predictors* of waveform occurrence. A neural network is a set of *neurons*, i.e. elements which are characterized at any time *t* by the property of firing or not firing. The Neurons are connected by *synapses*: each neuron fires or does not fires in according with operations performed on the set of its synaptic inputs. For instance, in the simplest models, the neuron performs a weighted average of the inputs, which may be also negative (inhibitory synapses) and applies a threshold. *Recurrent networks* are networks including feedback. It has been found that the introduction of adaptive time constants associated to each neuron of a recurrent network can improve the dynamical features of neural models.

Interpolation of a signal can be useful. Polynomial interpolation is often applied. Also the Fourier transform can provide efficacious interpolation, simply by first transforming, then performing some operations in the frequency domain (such as zero-padding or filtering), and at last inverse transforming. It can be shown that, if a signal *s(t)* has been sampled with sampling rate f_s according to the Nyquist-Shannon sampling theorem (thus providing a sequence *s[n]*), it can be correctly fitted according to the following formula:

$$s(t) = \Sigma_n \, s[n] \, Sa \, [\pi \, f_s \, (t-nT)],$$

where the function *Sa (x) = (sin x / x)* is called *sampling function* just for this basic property.

Interpolation can be important in order to characterize a signal with a small number of significant parameters. The same result can be obtained by simple models of the signal generator: for instance, ARMA models are often applied. These models aim at providing a better understanding of the properties of a time series and, possibly, predicting its future values. An ARMA model consists of two parts: an AutoRegressive part (AR), which is essentially given by an IIR filter described by a certain number of parameters, and a Moving-Average part (MA). The formula for a model with *p* autoregressive terms and *q* moving average terms for a signal *s(t)* is the following:

$$s(t) = \Sigma_{1 \le i \le p} \, \alpha_i \, s(t-i) + \Sigma_{1 \le i \le q} \, \beta_i \, \varepsilon \, (t-i) + \varepsilon(t),$$

where the α_i and β_i are parameters characterizing the model and $\varepsilon(t)$ is an error term which is generally assumed to be a random variable sampled from a Gaussian distribution with zero mean.

Data compression is becoming more and more important in signal processing, since the remarkable advances of technology make it possible to record huge amounts of data: for storage and transmission of this data, Compression (i.e. reduction in the number of bits) is generally useful if not necessary. We can distinguish lossy and lossless compression: for a number of purposes, lossy compression is sufficient: a recent remarkable example is given by the popular digital audio encoding MP3. The MP3 algorithms apply psychoacoustic models to discard components less audible to human hearing and efficiently encode the remaining information: the approach is similar to the lossy image compression format JPEG.

For signals characterized by multi-channel recording and approximately periodic behavior, such as ECG, there are three kinds of compression:
(a) intra-period and intra-channel (which is the simplest and can be carried out both in the time and in the frequency domain),
(b) inter-beat and intra-channel (which exploits the approximately periodic behavior,
(c) inter-channel (based on a comparison between the signals derived by the various channels).

In addition to non-linear operations that are very common in signal processing, such as thresholding and rectifying, other non-linear methods have been proposed based on the theory of deterministic chaos. Among the parameters that are related to models of this kind, the *Lyapunov exponents* have often been calculated. In dynamic systems presenting deterministic chaos, there is a remarkable dependence of the trajectory on the initial conditions. The distances among these trajectories are measured in the phase space: at the beginning of the motion, these distances increase approximately exponentially, and the rate of this divergence can be measured by the largest Lyapunov exponent. Lyapunov coefficients can be used to give an estimate of the rate of entropy production (i.e. non-reversible behavior) of the dynamical system. Chaotic dynamical systems are often associated with fractals, because objects in the phase space of these systems can be fractals, i.e. objects presenting a geometric shape that can be subdivided in parts, each of which appears as a scale-reduced copy of the whole. The *fractal dimension* indicates how completely a fractal appears to fill space, if the scale is more and more reduced. Lyapunov coefficients can provide an estimate of the fractal dimension of a non-linear dynamical system.

Modelling the signal generator as a dynamical system that consists of a small number of nonlinear sources has lead to the application of Independent Component Analysis (ICA), in order to separate signal components that could be attributed to the different sources. In particular, ICA has been applied for blind source separation or artifact removal. Figure 2.23 has been taken from an example available in the web site of the Institute of Technology of Helsinki. The two upper graphs show the original signals: a 5-fold application of the fast algorithm called

FastICA shows that the source signals are impulsive noise and a sinusoid (the third and fourth graph, respectively).

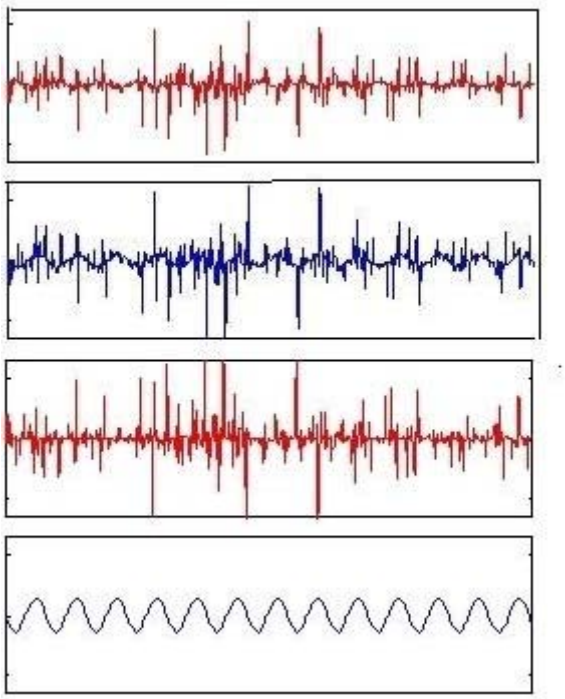


Figure 2.23. *Example of Independent Component Analysis (from the Web site of the Helsinki Institute of Technology*)

Independent Component Analysis is closely connected to Principal Component Analysis (PCA, also called Karhunen-Loève transform). Differently from the Fourier and the Wavelet Transforms, PCA has basis vectors that are not fixed, but depend on the data set. The original data are transformed to a new coordinate system; the first component retains the data-set properties that contribute most to the data variance; and the other components are calculated according to the same criterion. For this reason, PCA is a powerful tool for data reduction and, thus, feature extraction. If the data is represented by a matrix X whose rows gives the various features measured during an experiment that is repeated N times (N is the number of columns), then the PCA provides the following matrix:

$$Y = W^T X = \Sigma V^T,$$

where $W \Sigma V^T$ is the *singular value decomposition* of X^T, an important method for matrix factorization.

A crucial stage of signal processing is given by *feature extraction*, i.e. the identification of the features that, among those computed by the various methods applied, best characterize the signal. Indeed, the extracted features are not

necessarily a subset of the original features: they can be linear or non-linear combinations of them: statistical methods such as principal component and *factor analysis* provide these new features. The calculated features can be the input for further analyses. For instance, a Fourier analysis of the features describing an approximately periodic waveform can quantitatively assess its time variations. For instance, this kind of analysis can be applied to the study of clinically significant variations in the morphology of the T wave of ECG.

The selected features are generally exploited for signal *classification* and *clustering*, operations that consist in assigning each of the data to a class: this should be viewed as the final (or almost final) stage of most methods of signal processing. It should be underlined that this stage can only be performed if a previous analysis of the original signals is carefully carried out and significant features are calculated. Classification and clustering are basic in a number of fields, such as statistics, data mining, and artificial intelligence. The difference between classification and clustering is that for the former the classes are given previously (for instance, in biomedical research, they can be given by groups of subjects that are physiologically or clinically different), and for the latter the classes are somehow "discovered" by the application of the method. The phrases "supervised learning" and "unsupervised learning" are often used as synonyms of classification and clustering, respectively. For this stage, methods based on neural circuits, genetic algorithms, and syntactic algorithms can be applied.

2.2.2 Principles of ECG processing

The deliverable D9 provides the specification of all biomedical data, signs and symptoms relevant to heart failure. Apart from other mostly phenotypical or clinical HF biomarkers for which the extraction of the significant features is usually obtained (when raw data is available) with simple signal processing, raw ECG signal processing is usually quite complex, and is used in routine workflows, playing thus a significant and challenging role.

However, the information acquired from the ECG devices used in the HEARTFAID validation sites has not been established yet due to the lack of data format standardization and to the available information from the manufacturers for transferring the examination from the medical device to a host computer.

Thus, still we do not know if raw data and/or additional information (obtained by some processing performed on the medical device) will be available. For this reason we have decided to realize some multipurpose signal processing that can be useful for different medical devices. It provides some basic processing that is usually common to different modalities and at least substantially it helps the clinicians in the extraction of the significant features for the related modality.

A complete ECG processing chain includes ECG pre-processing and filtering, QRS detection and QRS classification (with normal beat selection). Once we have information on the normal and abnormal beats it will be quite easy to identify all the R-R intervals (between consecutive normal beats) from which the HRV variability in the time or frequency domain can be performed (Jekova and Krasteva, 2004; Tikannen, 1999).

At the same time the identification of the normal beat can allow (in short-term ECGs) the beat averaging and the identification of a reference beat on which all the significant measurements can be evaluated (peak of waves, intervals, etc.). These measurements are the base for the formulation of a diagnostic proposal. Analysis of the variability of the heart rhythm (heart rate variability or HRV) considers intervals between normal heart beats only and provides information about the functioning of the autonomic nervous system. Loss of HRV is pathologic and identifies, among others, cardiac and pulmonary patients at increased risk of mortality and diabetics with neuropathy.
Therefore, changes in HRV in response to stressors (e.g., tilt table testing, autonomic medications) provide insights into autonomic balance and adaptations. In addition, graphical displays of the rhythm of the heart identify patients with sleep apnoea and other forms of sleep-disordered breathing from routine Holter recordings. Such processing may include graphical and statistical data analysis using a variety of methods including linear ones, like the fast Fourier transform and autoregressive modelling or other alternatives based on information and nonlinear dynamics measures.
Precise QRS detection is the vital part of HRV. To facilitate the detection step an initial pre-processing stage is usually performed.

ECG pre-processing and filtering
Typical ECG pre-processing incorporates digital filtering in order to make the signal more readable and grant a better PQRST wave identification (Clifford, 2002; Chandra *et al.*, 2003).

The goals are the following:

i) Cancellation of the mains component (50 Hz in Europe and 60 Hz in US)
ii) Reduction of the baseline wandering (low frequency noise)
iii) Reduction of the high frequency noise
iv) Reduction of muscle or other artefacts (improper spikes, baseline jumps or saturations)
v) Noise estimation in each channel (to exclude the noisy channel from the processing)
vi) Identification of periods with specific rhythms (i.e. in case of a period of ventricular fibrillation, QRS detection has not to be performed in all that period)

QRS detection
QRS detection (Kohler *et al.*,2002) is the first basic step of any ECG signal processing (usually performed in the time domain) because it allows the detection of ventricular and supra-ventricular arrhythmias and it is the base for beat classification and averaging. The most common approach is to analyze the spatial velocity function (a further way to improve the signal-to-noise ratio because in 3 orthogonal channels the noise is uncorrelated while the signal is not). In general, the QRS complex is recognized from its slope, width and duration (Fernantez *et*

al., 2003). However, QRS detection can be performed also in a transformed domain using Hilbert transform or wavelet transform.

QRS classification and rhythm classification

A final crucial step after detecting QRS complexes is their classification by identifying if they are of the same type or if we have some spurious beat (rhythm anomaly) (Forbes, 1995). Beat classification is usually performed in the time domain with cross-correlation algorithms. Identification of premature atrial beats and period of ventricular fibrillation (Fernantez *et al.*, 2003; Amann *et al.*, 2005; Jekova *et al.*, 2004) is also important in order to select only consecutive normal beats as a starting point for the heart rate variability. Furthermore, having identified the normal beat we can, then, overlap and average all the normal beats (again for improving the signal-to-noise ratio) obtaining an averaged normal beat. This operation allows for a better estimation of the main morphological features (measurements) of the ECG (amplitude and duration of waves, intervals, etc.)

Heart Rate Variability

Once the R-R intervals for all consecutive normal beats are available, HRV is commonly performed in the time-domain and/or in the frequency domain.
HRV currently available methodologies are discussed below (see ESC-NASPE, 1996).

- Time Domain

Time domain linear analysis techniques are based largely on quantifying the variability in the R-R interval time series derived from an ECG recording. HRV can be quantified and statistically summarised using conveniently calculated measures such as SDNN (standard deviation of time interval between consecutive R peaks resulting from sinus node depolarisation in normal beats) and RMSSD (root-mean square of the difference between two adjacent R-R intervals in normal beats).

- Frequency Domain

Frequency domain represents the spectral alternative of the time domain. Hence, instead of just assessing the magnitude of the temporal variations in the cardiac rhythm, the frequency domain analysis provides the spectral composition of these variations. Using traditionally efficient algorithms such as the Fast-Fourier Transform, one is able to decompose the R-R temporal series (in consecutive normal beats) into its individual spectral components and estimate its Power Spectrum (Amann *et al.*, 2005). The latter can be averaged accordingly to define different frequency groups, i.e., very-low frequency (VLF), low frequency (LF) and high frequency (HF) bands. A variety of parameters (features) have been proposed to express the filtered components of the signal (ESC-NAPSE, 1996). The cumulative spectral power in the LF and HF bands and the ratio of these spectral powers (LF/HF) may form features that have demonstrable physiological relevance in healthy and disease states. Changes in the LF band spectral power (0.04– 0.15Hz frequency range) reflect a combination of sympathetic and

parasympathetic autonomic nervous system (ANS) outflow variations, while changes in the HF band spectral power (0.15–0.40Hz range) reflect vagal modulation of cardiac activity. The LF/HF power ratio is used as an index for assessing sympatho-vagal balance. However, it should be noted that such techniques require the signals to be 'time invariant' or 'stationary'. i.e., the individual spectral components do not change over the duration of signal acquisition.

- Time-Frequency Analysis

One step further stands the time-frequency analysis techniques capable of assessing the dynamic spectral properties of HRV. Such techniques are able to quantify, in a certain extent, both the time-varying characteristics of the autonomic influence on cardiac activity and the transient changes in heart rate response to these regulatory influences. Such algorithms are the short-time Fourier transform (STFT) or the Wavelet transform (WT) (Amann *et al.*, 2005), which have been proved to be of particular importance in investigating the transient episodes characteristic of various physiological and clinical states. However, even in this case STFT may be applied under the assumption of stationarity, whereas WT does not require it.

- Nonlinear Analysis

Another, totally different analysis approach makes use of the "state-space" reconstruction domain to investigate the non-linear complexities and deterministic nature of the control system regulating cardiac activity (Amann *et al.*, 2005; Jekova *et al.*, 2002). These non-linear methods include chaotic analysis, information domain measures such as approximate entropy, de-trended fluctuation analysis and heart rate turbulence analysis.

2.2.3 Principles of image processing

Digital Images and Colour Spaces

Consider the standard *grid* in the Euclidean *n*-dimensional space consisting of points having integer coordinates. Each point belonging to the grid is called a *voxel*. If the special case *n*=2, a voxel is called a *pixel*.

An *n*-dimensional *digital image* is a function defined on a finite subset of *n*-dimensional voxels and taking its values in a *colour space*.
For simplicity, consider a 2-dimensional rectangular digital image. It is effective to represent it as given by an *(M x N x C)* array *f (x, y, c)*, where *M* is the number of rows, *N* the number of columns of the image, and *C* is the number of colours.
We have *C = 1* for *monochrome images*, and usually *C = 3* for *colour images*. For RGB-coded colour images, *f (x, y, c)* represents the *intensity levels* of pixel *(x, y)*

for Red *(c = 1)*, Green *(c = 2)*, and Blue *(c = 3)*. In addition to RGB, a common format for colour images is YIQ: the *Y* component indicates the "Luminance" Level, thus providing a monochrome image, and the *I* and *Q* components indicate the "Chrominance Levels" for the orange-blue and the purple-green range, respectively. Another diffused format is HSI, where *H* indicates "hue", *S* "saturation", and *I* "intensity".

Image coding and image processing methods can be divided into two categories: actually, for a number of purposes the final result is required to be "objective", independent from the properties of the human visual system, while in other cases taking into account the characteristics of visual perception is helpful or even necessary. For instance, the YIQ system exploits the property of human colour-response of being more sensitive to the orange-blue range than to the purple-green: for this reason, the latter range requires less bandwidth in transmission systems. Another example is given by the HSI code: the basic reason for its large diffusion is that it describes colours in a way that is fairly close to human interpretation of colours: certainly, nobody would spontaneously describe a colour providing the respective percentages of red, green and blue, as the RGB code does. On the other hand, the RGB code, although being somewhat distant from human perception, is powerful for numerous important automatic analyses.

The values of the array representing a digital image can be integer (usually with 8, or 16, or 32-bit precision), or floating point numbers (with single or double precision), or logical. For instance, MATLAB provides eight numeric classes, one character class, and one logical class. Images are often *indexed images*: colours are stored in a *palette*, or lookup table: the colours available for are those and only those contained in the palette. In addition to standard palettes (such as Window, Macintosh, Netscape and "Web-safe"), it is easy to create adaptive palettes for specific images.

A number of processing methods focus on the colour properties of an image.

A typical example is given by *dithering*, i.e. using differently coloured adjacent pixel to simulate images and shades that do not exist in the colour palette. Dithering is a classic method for "fooling the eye" by taking advantage of the characteristics of visual perception. In spite of the importance of methods proper of colour images, in the following we will only consider monochrome images, for the reason that the components of a colour image can be formally considered as monochrome images; moreover, in significant cases, such as the *Y* (Luminance) component of a YIQ Image, they are exactly monochrome images.

Image Preprocessing

Image pre-processing refers to the task of preparing an image for further analysis. For example, it may be used to correct defects, to ameliorate the illumination conditions or normalize its intensity to use standardized automatic procedures. Image pre-processing may be seen as the application of a cascade of *image filters*, some common types of which are described in the following.

Intensity filters are basic tools for image processing. An example is given by *histogram equalization*.
For simplicity, we can consider an image whose intensity levels $g(x, y)$ ($1 \leq x \leq M$; $1 \leq y \leq N$) are 8-bit integers, i.e. are in the $[0, 255]$ range. Its histogram $h(i)$, where $0 \leq i \leq 255$, is the number of occurrences of Intensity Level i: of course, $\Sigma_i h(i) = MN$ (i.e. the total number of pixels). We can construct an Image $g'(x, y)$ such that

- its histogram $h'(i)$ is approximately uniform (generally, the histogram will not be exactly uniform, due to the fact that intensity levels are discrete);
- for each pixel pair, if $g(x, y) \leq g(\xi, \eta)$, then $g'(x, y) \leq g'(\xi, \eta)$.

Figure 2.24 shows an image of the heart obtained by MRI and the same image after histogram equalization. Figure 2.25 shows the intensity histogram of the original image.

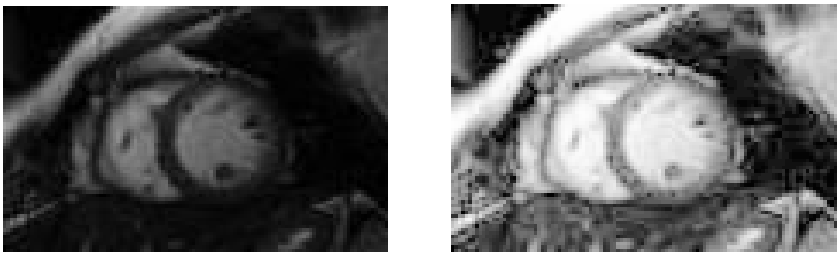


Figure 2.24. *Short axis MRI image of the heart. Left: original. Right: after histogram equalization*

Another basic tool is *neighbourhood processing*: the intensity value of each pixel is modified according to the values of the neighbour pixels. For example, a $K \times K$ *mask* $m(x, y)$ is often applied, and then *correlation* (a linear operation) between this Mask and the $K \times K$ sub-images obtained considering each pixel (x, y) and its (K^2-1) neighbours is computed. From the point of view of signal theory, this is an example of *linear filtering*, and the filter applied is a FIR (Finite Impulse Response) Filter.
As an instance of commonly applied mask, the so-called Sobel mask provides a simple method for approximating a vertical gradient. Some methods of neighbourhood processing are non-linear.

For example, a second-order combination of a Sobel mask and its transpose, followed by the application of an appropriate threshold, provides a simple preliminary method for edge detection. Figure 2.26 shows the result of the application of this method to the left image in Figure 2.24.

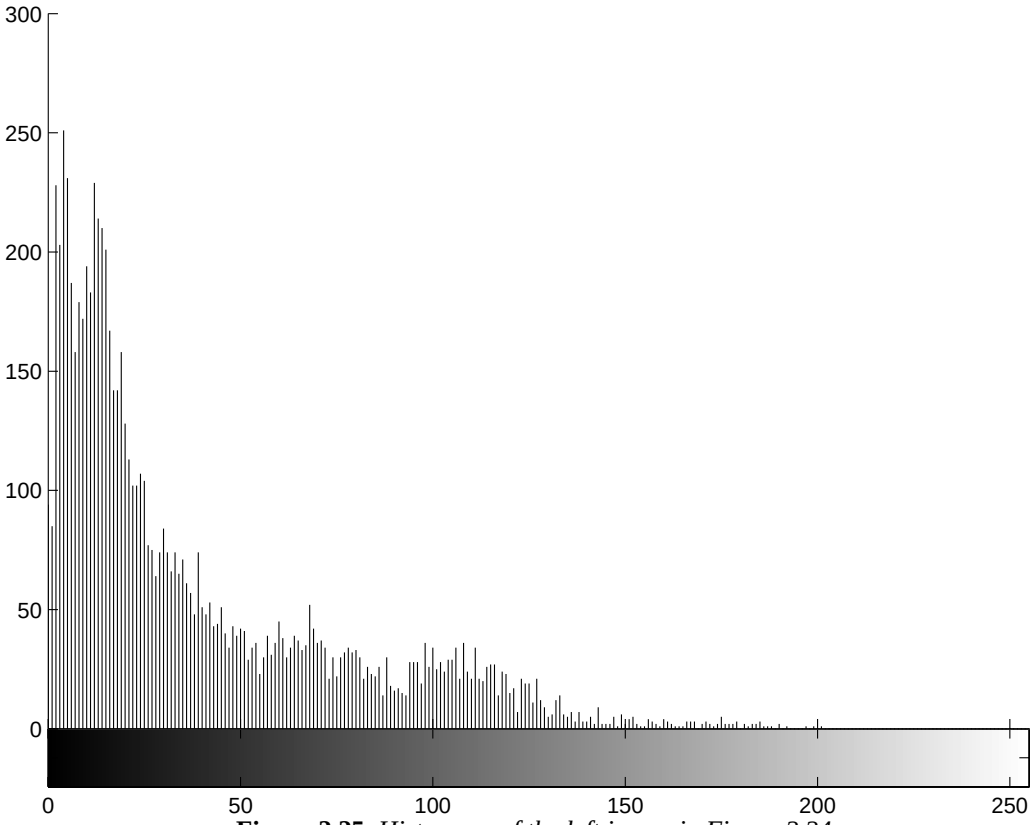


Figure 2.25. *Histogram of the left image in Figure 2.24*

Figure 2.26. *Sobel filter applied to the left image in Figure 2.24*

Other neighbourhood filters commonly applied include *mean* and *median filters*, *mathematical morphology filters* and the more general class of *voting filters*.

So far, we have considered examples of image processing in the space domain. In the same way as one-dimensional (1-D) signals in the time domain are often

processed in the frequency domain, images, which are 2-D signals in the space domain, can be processed in the domain of space frequencies. The 2-D Discrete Fourier transform (2D-DFT) is mathematically given by the successive application of two 1-D DFT. Calling *u* and *v* the space frequency (which in the following we will simply name "frequency") coordinates, the mathematically intermediate step given by the first application of the 1-D Fourier transform to an M x N Image *g(x, y)* gives:

$$G_1(x, v) = \Sigma_y\, g(x, y) \exp(-2i\pi vy / N),$$

where *i* is the imaginary unit.
After the second and final step, the Fourier transform of *g(x, y)* is given by

$$G(u, v) = \Sigma_x\, G_1(x, v) \exp(-2i\pi ux / M).$$

In the same way as a time signal can be filtered in the frequency domain, an image can be filtered by Fourier transforming, multiplying in the frequency domain, and at last inverse Fourier transforming. Generally, low-pass filtering "blurs" an Image, and high-pass filtering "sharpens" it. The linear method for neighbourhood processing above described can be easily applied in the frequency domain, simply multiplying by the Fourier Transform of the applied mask. This is an immediate consequence of the equivalence between *correlation* in one domain and *algebraic product* in the other.

The 2-D Fourier Transform is helpful for *noise reduction*. In fact, in a number of cases, noise presents periodicities that can be efficaciously detected by Fourier transforming and removed by filtering in the frequency domain.

Among the various methods used for noise reduction and image restoration, *Wiener filtering*, introduced by Wiener in 1942, has played a major role. It can be applied in the frequency domain. Let *F(u, v)* be the Fourier transform of an image (of which a degraded representation *G(u, v)* is available) and *N (u, v)* the squared module of the noise Fourier Transform.

The *signal-to-noise power ratio* is defined by:

$$\rho(u, v) = F(u, v)\, F^*(u, v) / N(u, v).$$

If H(u, v) is the so-called *degradation function*, the application of the Wiener filter provides an estimated restored image, whose expression in the frequency domain is:

$$\Phi(u, v) = H(u, v)\, H^*(u, v)\, G(u, v) / (H^2(u, v)\, H^*(u, v) + \rho(u, v)).$$

In addition to the Fourier transform, other transforms are useful for image processing. The Discrete Wavelet Transform (DWT) is often applied. While the expansion functions for the Fourier transforms are sinusoids (which have infinite "duration", using this term for space), the wavelets that provide the *kernels* of the

discrete wavelet functions have finite duration. Each *mother wavelet* is translated in order to give information in the space domain and is scaled in order to give information in the frequency domain. Different scales correspond to different decomposition levels of the original image. An important algorithm for the computation of the discrete wavelet transform is the *fast wavelet transform*, which is computed by means of iteration of two operations: *filtering* (both *high-pass* and *low-pass*) and *down-sampling* (i.e. picking every other point from a sequence of points). The calculation of the discrete wavelet transform coefficients (as well as the coefficients of other transforms) often allows characterizing an image by a reduced number of significant features, and thus obtaining efficacious representation and compression.

Image Segmentation

Image segmentation aims at separate the image domain into regions that are, in a broad sense, meaningful for an image analysis or interpretation task. For example, in cardiac imaging, the goal of segmentation could be to determine the region in an image corresponding to the left ventricular cavity. It is naïve to believe that this task could be accomplished solely on the base of grey-level information, since actually in many cases a priori information on the specific region to be segmented should be introduced.

Segmentation is an extensive topic, featuring an uncountable number of strategies which can be roughly categorized into three main classes:

1. Region based methods
2. Edge based methods
3. Classification methods

Loosely speaking, region-based methods try to segment a region on the basis of the value and uniformity of the intensity function of the image. Edge based methods look instead at the discontinuities of the intensity functions, which are understood as contours separating one region of interest from the background.

In classification methods, local features are computed in any image point and are subsequently classified by supervised or unsupervised artificial intelligence methods. These three basic strategies are in some way overlapping and many successful segmentation schemes use hybridization of them.

One of the most successful schemes for segmentation is active contours, introduced in the seminal paper by Kass *et al.* (1987). The basic idea is to insert a *deformable curve* in the image domain, that can move and stretch like a rubber band under the action of some *forces* derived from the image (*data-driven forces*). The data-driven forces are designed to make the contour evolving towards the boundary of the region of interest. Forces may include region based, edge based clues or both. Usually *internal forces* are included into the model and they act as a regularization factor to prevent the contour to break or bend abnormally (a surrogate of elastic force in a rubber band). Finally, a priori information may be introduced using *shape forces* that prevent the model to deform in a way that is

not normally observed in nature. Some of recent active contours schema (see e.g. Sethian, 1996; Osher and Paragios, 2003), based on advanced pure and applied mathematical stuff, gave superior segmentation capabilities in medical imaging.
It is impossible to get through the details of segmentation methods, so we prefer to describe the peculiarity encountered when analyzing the imaging modalities considered previously.

- US Segmentation

Segmentation of ultra-sound images is particularly difficult since this modality has several specificity which make automatic analysis procedures extremely challenging. Indeed US images are strongly anisotropic: the reflection intensity, the spatial resolution, the signal to noise ratio depend on the depth and the angle of incidence of the US beam. Further US images exhibit artefacts, like shadowing from the lungs and dropouts of structure parallel to the US beam. In echocardiography, there is also lower contrast between the endocardium, myocardium, epicardium w.r.t. MRI and minor spatial resolution. Of course, good contrast is important for robust segmentation.
Finally, speckle pattern is a typical phenomenon in US images due to the interference of waves backscattered from neighboring tissues, which add some complication to the problem.
For these reasons, border delineation in US images is considered more a problem in expert systems that a problem in image processing. Actually segmentation cannot be based on image evidence, but a priori information should be exploited. Besides, a good point is to perform a spatio-temporal analysis, i.e. try to segment globally a time-sequence of images. Indeed, a time-coherent segmentation may solve the usual ambiguities which one faces dealing a frame by a time and mimics the procedure followed by some expert observers.

A number of methods for segmentation of echocardiographic data have been recently introduced. Generally, it was found that commonly applied approaches to edge detection by means of gradients were not helpful (see Figure 2.27 for some edge detectors results). As an interesting example of single-frame methods, we can consider that proposed by (Mignotte and Meunier, 2001) for the analysis of short-axis parasternal images. They applied active contour algorithms and introduced suitable statistical hypotheses for determining the initial contour, for modelling the grey level distribution, and for minimizing the energy function. Another example is given by the work by (Chen *et al.*, 2003). They used prior profiles for region shape and for intensity and determined the region borders by means of an optimized balance between the actual image information and the prior estimations. Among the methods based on artificial neural networks, we can mention the two-layer back-propagation network applied by (Binder *et al.*, 1999) to parasternal short-axis images. An example of spatio-temporal approach is given by the method applied by (Chalana *et al.*, 1996) to short axis views.

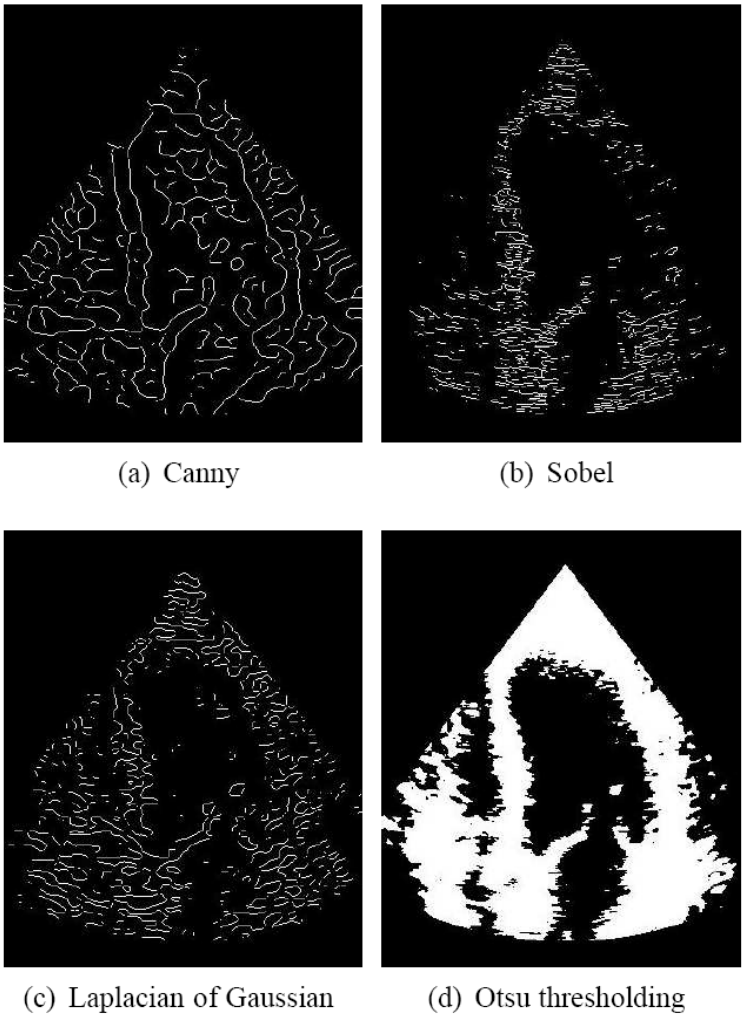


Figure 2.27. *Edge detector vs. thresholding in an apical view of the left ventricle*

This method focused on the motion between consecutive frames, which was described as due to an attracting force, which was quantitatively expressed by an external energy term. A modification of the Active Appearance Model (AAM) was proposed by (Bosch *et al.*, 2002) in order to obtain the motion of the endocardium. The latter method is exemplar in showing how sophisticated a segmentation strategy can be. An AAM is a technique of analysis by synthesis by which an anatomical structure and its deformations can be studied. More precisely, it tries to encode an average anatomical structure and the deformation of it that show up more commonly in nature. After having built such a model, it can be used for segmentation tasks. Actually segmentation boils down to fitting the model to the particular anatomy under examination. In (Bosch *et al.*, 2002) this strategy is extended to deal with 2D+1 data, namely to study the full deformation cycle of a 2D view of the left ventricle. In particular they introduce an *average heart beat,* shown in Figure 2.28 together with the deformation along one of the more common 'deviation' direction (actually, this is a direction obtain by PCA on

a training set of heart beats). It is important to note that this method allows for segmentation enforced by a priori knowledge (a shape prior obtained by learning) and spatio-temporal coherence.
For a more complete review, we refer to the recent careful survey of methods for echocardiographic image segmentation proposed by Noble and Boukerroui (2006).

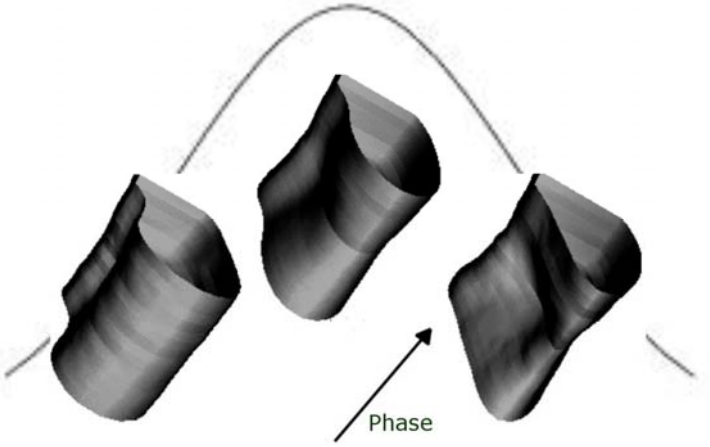


Figure 2.28. *Average heart beat (at the center) and one of its principal deformation. (modified after Bosch et al.,2002)*

- MRI segmentation

Magnetic Resonance Imaging has proved to be more reliable than other techniques, both in supplying accurate morphological information and in assessing heart functions. In general, the segmentation problem in MRI is considered easier than the corresponding one in US. In particular the delineation of endocardium (roughly speaking, the inner border of the left ventricle) seems to be well-understood.
However, due to noise or acquisition artefacts, visual information can be corrupted or ill defined. Moreover, in MR images, the presence of *papillary muscles* and *trabeculations* causes problems in detecting the LV contours. In such cases, only expert knowledge can help: the exact location of the contours cannot be based only on image evidence, but should be learned from examples provided by expert observers.

Figure 2.29. *Endocardial (left) and epicardial (right) borders in a short axis MRI*

As for US, usually, researchers have tried to design *ad hoc* algorithms able to incorporate *a priori* information about the LV shape. Model based surface detector have been widely used: for example, (Declerck *et al.*, 1997) employed a *Canny-Deriche* edge detector in a 3D polar map to segment endocardial and epicardial surfaces, while (Faber *et al.*, 1991) defined a hybrid spherical-cylindrical coordinate system.

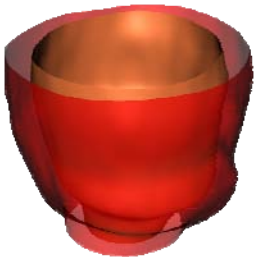


Figure 2.30. *Reconstruction of the left ventricle from a multislice MR image*

Active contours have been used for cardiac MRI segmentation for long time. Recent improvements in this field include works by (Jolly *et al.*, 2001) who reduced sensitivity to initial contour through Dijkstra algorithm, and by (Paragios, 2002) and (Huang *et al.*, 2004) who introduced deformable models influenced by forces derived from image region information. A biomechanical volumetric model of the heart, and not only of its boundary surfaces, is proposed in (Sermesant *et al.*, 2003) where segmentation of time series of images is obtained by means of a non-rigid deformation procedure. Active Appearance Models (the ones described in some detail in previous section) have been used also to segment time-series of images (Mitchell *et al.*, 2002).

Finally, also neural networks approaches have been proposed. In (Stadilis *et al.*, 1999) a *Generating-Shrinking Neural Classifier* is used to distinguish among lung, blood and myocardium points. This classification allows to extract a set of points on myocardial surfaces and, then, to assess parameters for a wavelets-based model.

Besides the great number of bibliographic references, however, no method seems to have achieved at the same time robustness, adaptability and good automation in the task of inner and outer border delineation in cardiac MRI.

- Chest X-ray segmentation

One of the first steps in many pulmonary analysis tasks is the delineation of the lung and heart boundaries from image data obtained by chest X-ray. Usually chest X-ray film are manually analyzed and the number of recent medical imaging works in the field is small when compared to cardiac image segmentation in MRI and US. However, some automatic methods have been described. For example (Yue *et al.*, 1995) introduced a technique for ribs and cage identification based on polynomial curves fitting and dynamic programming. (Brown *et al.*, 1995)

employed instead a semantic network of thoracic anatomy (including relationships among the different structures) to fully segment chest X-ray film. The approach was based on edge detection and subsequent model fitting. A neural network approach is described instead in (McNitt-Gray *et al.*, 1995), where a classifier is trained over small regions to distinguish among heart, lungs, arms and background. Such an approach is sufficient to determine cardiothoracic ratio, the most relevant quantitative parameter derived from chest X-ray in heart failure.

Image Registration

Image registration represents a basilar problem in medical imaging. Indeed often diagnosis, analysis and models need as a prerequisite the comparison or fusion of images acquired on the same object or on objects belonging to some class. Image registration, whose aim is to determine a geometric transformation to align the points in the various images, define as the comparison should be performed.

For example, if a pair of images depicts the same object, a typical application is to look for a geometric transformation that align the points in the two images corresponding to same *material* point in the real world. If the images depict instead two different anatomical structures but belonging to the same class (say for example the femur of two patients), image registration may be useful to *match* structure of interest visible in both images and peculiar of the structure under examination. More precisely, let f,g be a pair of images. The aim of registration is to find a geometric transformation T between their domains (i.e. their respective set of voxels), called a *matching*, in such a way that an objective function $D(f,g,T)$ expressing the goodness of the alignment is maximized.

Thus, in the theory of registration, crucial issues are the design of suitable functionals $D(f,g,T)$ and the development of techniques for their optimization.

In (Mainz and Viergever, 1998) an accurate categorization of registration algorithm is provided. In particular, when choosing a matching strategy, one should consider:

1. A *deformation model*, that is a soundly mathematical formulation of the generic geometric transformation used to align a pair of images. The choice depends on the a priori knowledge on the form of the expected alignment. For example, the misalignment between two images may be due to object deformations, spatial distortions, or merely to a change of the reference system. Usual deformation models range from low dimensional transformation groups such as rigid and affine transformations to Free Form Deformations (FFDs) which can represent arbitrary diffeormorphism of the image domain.
2. The *registration basis*, which consists in the choice of the features to be used as a clue to find the optimal alignment. Namely these are the image features to be included in the functional $D(f,g,T)$. For example, one may require correspondence among landmarks individuated in both images or require the *edge maps* (obtained by some edge detector) to overlap tightly. As an example, as an objective functional one may consider the distance between the geometric structures delineated in the pair of images. The method will be then called *Geometric Feature Based* (GFB) (see Figure

2.31). In some other cases, if the intensity functions in the images is related to the properties of the depicted tissue, a notion of similarity of the intensity function may be used (for example the *sum of square of pixels differences, mutual information, correlation ratio,…*). The method will then be called *Standard Intensity Based* (SIB).

3. *Modality* refers to the technology by which the images are acquired (e.g. US, MRI, CT,…). *Monomodal* registration denotes registration between two images acquired by the same technology, while *multimodal* is used for registration between two images obtained by different technologies (e.g. CT-PET).
4. In relation to the *subject*, one finally distinguishes among *intra-*, *inter-patient registration* and *atlas.*

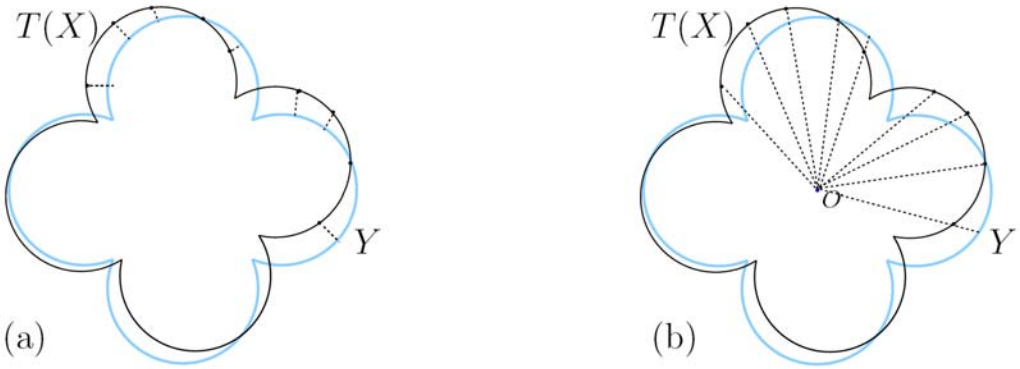


Figure 2.31. *Two examples of registration GFB. In (a) a point in the darker curve finds as homologous the closest point in the lighter curve. In (b) the homologous point lays instead on the ray starting from the centroid of the lighter curve. In literature, these two methods give rise respectively to the iterative closed point and head and hat algorithms.*

- Cardiac Image Registration

Registration of cardiac images is usually performed in order to:

- Fuse in a global picture information about the heart gathered from various modalities (e.g. relate heart perfusion to the anatomical structures delineated by means of MRI).
- Explore by means of registration algorithms the *motion* of the cardiac structures.
- Construct an *average heart beat* by means of inter-patient registration, so as to encode a *prior* for other image analysis tasks (e.g. construct a *shape prior* term to enforce cardiac image segmentation).

The literature of the subject is extensive and we refer to (Makela *et al.*, 2001) for a careful review. We just recall that GFB methods in cardiac analysis are difficult to use; actually on the heart there are not sufficiently many well defined landmarks; one should use thus GFB methods starting directing with contour lines or surfaces.

Far sophisticated techniques for surface matching based on curvatures and its extremal point have been presented (see e.g. Pennec *et al.*, 2000). SIB used in combination with FFDs are instead promising; but they cannot be used with success in US images, since the intensity function does not reflect well defined physical properties of the imaged tissue.

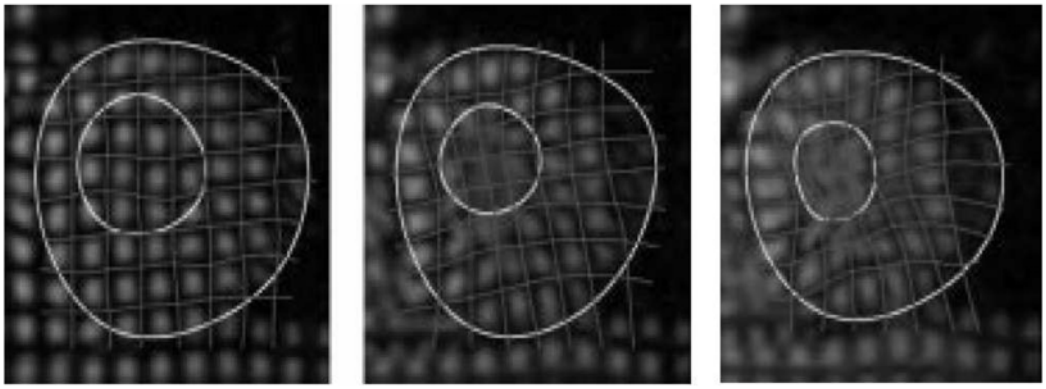


Figure 2.32. *Analysis of myocardial motion by means of TAGGED MRI. In (a), taken at end-diastole, a grid pattern induced by tissue magnetization is clearly discernible. The grid then deforms during the heart beat. Registering the grids enables cardiac motion analysis (modified after Amini et al., 1998)*

Several attempts to *motion estimation* have been reported. The most commonly used employ *optical flow* principle or some related algorithm (for example the so-called *Demons algorithm* (Thirion, 1995)). Good results were obtained using TAGGED MRI as imaging modality (see Figure 2.32).

2.3 Open problems: instrumentation lacks and intra-/inter-observer variability

The power of the discussed diagnostic techniques in the management of heart failure is invaluable. Often the devices provide semi-automated procedures for quantitative measurements that can be readily transferred to the platform, without requiring any further signals and images processing. However, despite the capabilities of proprietary software packages in bundle with new devices, the role of data processing for the methodologies relevant to HF is still crucial also in the routine clinical environment. Actually some measurements exhibit strong intra- and inter- observer variability, which, although it doesn't alter their clinical validity, may hide some peculiar information. Considering that these parameters will be eventually processed by data mining procedures and used in the CDSS, higher precision is desirable. Thus, signals and images processing should try to produce automatic and semi-automatic methods to lead to standardized parameters for the platform. Those software pieces, integrated with a suitable end-user interface, will eventually be able to cope with software shortage in older

devices, thus extending the potential application field of HEARTFAID platform to poorly equipped medical premises.

As an example we consider echocardiography, being it the most relevant and wide spread imaging diagnostic modality in heart failure, and in particular to 2D echo. Such technique allows for the estimation of heart chambers sizes and volumes, obtained by manual tracing of their border. Since however 3D structures are incompletely characterized by 2D views, formulas based on geometric assumptions about the shape of the chamber are commonly applied. The validity of the performed measurements depends on several factors. Actually misshapen heart (frequently encountered in heart failure patients) is not correctly represented by the geometrical assumptions used in the computation. In second place, the identification of the correct view is crucial, since a transducer ill-positioned may lead to volume overestimation. Studies report that in 68% of examinations, transducer is >5% far from its ideal position and that the image plane angles vary by 15 degrees in 52% of cases, even when the examination is performed by expert sonographers. However, it is impossible to deal with wrong view selection in post processing, unless one employs an acquisition protocol based on multiple image planes, as suggested by the American Society of Echocardiography in 1989. Finally, manual contour tracing is extremely prone to intra- and inter- observer variability, also in relation with the specificity of US images. For example for left ventricle volume at end-diastole the standard error is of about 20ml (in vivo studies) and 17ml at end-systole. For comparison, it has been reported that by contrast ventriculography and 3D echo the standard errors are instead 8.3 ml and 11ml, respectively.
Considered that the variability in border tracing is not expected to affect too much a volume (since deviations in various regions may compensate for one another), these standard errors are not encouraging. Even worse, when considering ejection fraction or fractional area change, measures instability is stronger.
Indeed, from a mathematical point of view, these parameters are a sort of first order derivatives, and so they lay in the category of numerically instable computations known as *catastrophic cancellation*. For example, fractional area change may exhibit occasionally a variability of 15% on the same image sequence of the same patient. Uniform error in border tracing of 1mm can lead to a fractional area change ranging from 42% to 58%.
Computation of wall thickness and thickening is even more prone to error. Usually it is based on linear measurements whose lengths are of the order of 20 pixels. Thus, a 2 pixels error in border tracing produces an error of about 10% in wall thickness.
For these reasons, variability is not just a *theoretical problem* but does have serious impact.
Some other echo-provided parameters are clearly of semi-quantitative nature, for example the segmental scoring of wall motion and thickening. In this procedure, a score is given to each *segment* of a standardized subdivision of the left ventricle, depending on the strength and coordination of wall motion and thickening. The individual scores are summed up to produce a global score. Agreement among

different expert observers for the global score is lower at 62-88%, while exact agreement on individual scores is seen only in 44% of the segments. Quantitative wall motion analysis is thus seen as a further activity. Since wall motion is subject to artifacts introduced by the translational motion of the heart within the chest, *registration procedure*s should be employed to factor out this global motion, irrelevant in the analysis of the intrinsic dynamics.

When considering one-dimensional biomedical signs, a big problem is represented by the *lack of standards*, that could make more difficult the adoption of signal processing techniques in the HEATFAID platform. Indeed, while the imaging devices have a largely diffuse standard for data format and communication protocol like DICOM, unfortunately the non-imaging devices do not have a similar standard largely adopted by manufacturer.
This lack of adoption of interoperability standards from manufacturers is the main cause of the typical heterogeneous scenario we have to face as far as the medical devices concern.
Assumed that the medical device used for a specific examination is able to transfer the acquired examination to third-party software running on a computer (this is not always true especially for low-cost or old models medical devices), we can classify the medical devices according to the following categories:

- Medical device without diagnostic capability (raw data device)
- Medical device with diagnostic capabilities (i.e. perform measurements, extract features from the raw data) (i.e. diagnostic resting ECG devices, Holter ECG, etc.).

The second classification criteria are:

- Is the device able to export the raw data in an open/disclosed format?
- Is the device able to export the measurements performed by the software running on the medical device itself?

Specifically, diagnostic resting ECG devices are usually able to transfer raw data, measurements and diagnostic proposal to a computer even if very often the format used is not open/disclosed but proprietary. The reading station of Holter ECG devices is usually able to export examinations (raw data, summary report, results of HRV, etc.) but also in this case very often the format (especially for raw data) is not open/disclosed.
In this scenario, the approach that has to be used for each device (brand/model) of potential interest for the HEARTFAID platform is to analyze which information can be exported with functional and semantic interoperability to the HEARTFAID platform.
The answer can be:

- None
- Only raw data
- Only some measurements (some processing/editing results)
- Both raw data and measurements

Thus, the signal processing that can be consequently applied is strongly influenced by the exporting capabilities (with functional and semantic interoperability) of each medical device.

We can identify the following categories for raw data devices:

- Raw data devices without the capability of exporting the data
  - No signal processing can be performed. The clinicians have to manually evaluate and insert the features/parameters in the patient clinical record.
- Raw data devices with export in open/standard format
  - Signal processing can be performed in order to evaluate the features/parameters of interest. Signal resolution, duration and quality have to be sufficient for the implementation of reliable algorithms. Still there is the challenge of identifying new features/parameters that can be extracted from the raw data (technical/clinical research)
- Raw data devices with export in proprietary format

  In this case raw data can be available if one of the following conditions can be applied:
  - The release of technical information by the manufacturer (usually under signing a NDA) (time-consuming)
  - 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 (money-consuming)

  The last approach is not affordable and can be taken into account only for particular devices of strategic importance.

And the following categories for diagnostic devices (devices with capability of evaluating measurements/features) (this category seems to be more interesting due to the fact that most medical devices resulting from the inventory in the validation sites belong to this category):

- Devices with export in proprietary format.

  In this case raw data and/or measurements can be available if one of the following conditions can be applied:
  - The release of technical information by the manufacturer (usually under signing a NDA) (time-consuming)
  - 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 (money-consuming)

  The last approach is not affordable and can be taken into account only for particular devices of strategic importance.
- Devices with diagnostic capability and without the capability of exporting the data (raw data and extracted features/parameters)
  - No signal processing can be performed. There are some studies in order to reconstruct the raw data from the scanning of a print-out (ECG), but the quality of the reconstructed signal cannot be enough for a proper diagnosis, thus the extraction of further features very likely will not be reliable. In this case, the clinicians

have to manually insert the features/parameters already evaluated and displayed/printed by the device in the patient clinical record.

- Devices with diagnostic capability and the capability of exporting the data (raw data and extracted features/parameters) in an open/standard format
  - It does not make any sense to re-evaluate the same features or the same parameters already evaluated by the device itself (very likely with more consolidate and performing algorithms). In this case we should simply identify if there are additional features of clinical interest that can be extracted from the raw data for technical/clinical research.
- Device with diagnostic capability and the capability of exporting the data (only extracted features/parameters) in an open/standard format
  - No signal processing can be performed. There are some studies in order to reconstruct the raw data from the scanning of a print-out (ECG), but the quality of the reconstructed signal cannot be enough for a proper diagnosis, thus the extraction of further features very likely will not be reliable.
- Device with diagnostic capability and the capability of exporting the data (only raw data) in an open/standard format
  - Signal processing can be performed but the features/parameters already evaluated and displayed/printed by the device cannot be better evaluated (very likely with more consolidate and performing algorithms). The only purpose of such reevaluation might be the avoidance for the clinicians of manually inserting such features/parameters in the patient clinical record (but the algorithms have to be reliable). Still there is the challenge of identifying new features/parameters that can be extracted from the raw data (technical/clinical research).

Further difficulties may arise due to the fact that only some partial results from the internal signal processing of the medical devices (without any raw data) are available to the HEARTFAID platform in an interoperable format.

This situation is likely to happen with some Holter ECG device resulting from the inventory performed during WP2. Some device does not export the raw data in an intelligible format but exports an ASCII file with all the timestamps of the QRS detected and edited at the review station. Unfortunately there is no label associated with each QRS and thus it is not possible to reconstruct the morphology of each detected QRS and to state if it is a normal QRS or not. This problem might hamper the identification of all R-R intervals between consecutive normal beats and have a negative impact in the HRV analysis. Of course in this case specific algorithm can be applied to the R-R sequence in order to identify rhythm anomalies of non-morphological nature (premature atrial beats, bigeminy, trigeminy, etc.) or to eliminate potential R-R artifacts but their full validity have to be demonstrated.

3. Methodological Foundation of Decision Support Systems

The concept of Decision Support System (DSS) is extremely broad, because of the many available approaches to decision-making and of the wide range of domains in which decisions are made.

The need for computerized mechanism for decision support comes from well-known limits of human knowledge-processing. More than forty years ago, Simon introduced the notion of *'bounded rationality'* of human decision-making abilities [-]. This notion argued that people are limited in the amount of information they can process and the methods they use to integrate information: a person's capacity for processing the contents of his or her immediate field of awareness is limited to manipulating up to about seven pieces of knowledge at any one time. The stress, errors and oversights that can result from being overloaded with knowledge can be just as detrimental as not having enough knowledge. In addition, a person may not be especially skilled at some kinds of knowledge manipulations (e.g., mathematical ones). More in general, it has been noticed that the need for support for human decision makers is due to four kinds of limits: cognitive limits, economic limits, time limits and competitive demands [Holsapple and Whinston 1996]. Simon believed that if these limitations could be transcended, then the decision-making effectiveness would be enhanced. Computers are known for their amazing power to organize, store, process, and retrieve vast volume of information. The integration of human and the computer was foreseen hence appears as the best method for achieving new heights for decision makers.

Researchers and technologists have built and investigated DSS for more than forty years: the history of such systems begins in about 1965, starting with the development of *model-based* DSS. The concept of DSS became an area of research of its own in the middle of the 1970s, before gaining in intensity during the 1980s. In the middle and late 1980s, *Executive Information Systems* (EIS), *Group Decision Support Systems* (GDSS), and *Organizational Decision Support Systems* (ODSS) evolved from the single user and model-oriented DSS. At the beginning of 1990s, DSS knew a huge spread thanks to *Data Warehousing* and *On-Line Analytical Processing* (OLAP). As the turn of the millennium approached, new Web-based analytical applications were introduced.

The methodological foundations of DSS are multidisciplinary, including (but not exclusively) Simulation Methods, Software Engineering, Database Management, Artificial Intelligence, Computational Intelligence, Image/Signal Processing and Analysis, Human-Computer Interaction, and Telecommunications.

In particular, the last forty years witnessed a parallel evolution, within the field of Artificial Intelligence, of the so-called *Expert Systems* (EE) or *Intelligent Decision Support Systems* (IDSS). As we will see in the following, they can be considered a special class of DSS, designed to formally encode and allow the use of the specific knowledge of an *expert*, i.e. a person or a group of persons having high-

quality experience and expertise (this is also the reason why they are often called *Knowledge-Based Systems* or *Knowledge-driven DSS*).
Expert Systems are the commonest type of *Clinical DSS* (CDSS), that is DSS used in routine clinical environment. They contain medical knowledge about a very specifically defined task and are used to to improve the quality of care and/or reduce cost without loss of quality.

In this chapter, we provide a brief introduction to Decision Support Systems, in relation to the decision theory. Moreover typical DSS models and structures are presented. A special section is dedicated to Expert Systems, and related issues such as knowledge representation and inference engine modelling. Application to the clinical field is widely discussed in the section on CDSS.
Important issues regarding DSS design and success factors are examined in the last section.

3.1 Decision Support Systems

Several definitions of Decision Support Systems have been presented in the literature, placing different emphasis upon their conceptual and technological aspects. We can say that **a *DSS* is a computerized system which assists *semi-structured* or *unstructured* decision making by combining comprehensive analysis and exploration of current and historical data, sophisticated analytical models and tools, and user-friendly software into a powerful system than can guide the decision makers towards more effective and efficient choices**. Such definition asserts that a DSS can be considered any system that aids one or more people in making decisions, but some important characteristics should be fulfilled. In deed, according to Sprague & Carlson (Sprague and Carson, 1982), a DSS should

- be aimed at the less well structured, underspecified problem that decision makers typically face;
- attempt to combine the use of models or analytic techniques with traditional data access and retrieval functions;
- specifically focus on features that make it easy to use by “non-computer” people in an interactive mode;
- emphasize flexibility and adaptability to accommodate changes in the environment and decision-making approach of the user.

In order to understand better what this definition means and go further in the discussion of decision support, it is necessary to have a perspective on the nature of the decision process and, thus, what it means to support it. According to Simon (Simon, 1977), decision making consists of three stages:

- *Intelligence*: searching the environment for conditions calling for decisions; raw data are collected, processed, and examined for clues that may identify problems

- *Design*: investing, developing, and analyzing possible courses of action. This involves processes to understand the problem, generate solutions, and test solutions or feasibility.
- *Choice*: selecting a particular course of action from those available. A choice is made and implemented.

This means that we can look at decisions in terms of three areas or components. The first component is the *data* collected by a decision maker to be used in making the decision. The second component is the *process* selected by the decision maker to combine this data. Finally, there is an *evaluation* or *learning* component that compares decisions and examines them to see if there is a need to change either the data being used or the process that combines the data. These components concur to define a possible categorization of decisions on the basis of their degree of *structure*.

A *structured* decision is one in which all three components can be fairly well specified, i.e., the data, process, and evaluation are determined. Usually structured decisions are made regularly and therefore it makes sense to place a comparatively rigid framework around the decision and the people making it. An example of this type of decision may be the routine credit-granting decision made by many businesses. In addition the way in which the data is combined is likely to be consistent (for instance, household debt must be less than 25 percent of gross income). Finally, this decision can also be evaluated in a very structured way (specifically when the marginal cost of relaxing credit requirements equals the marginal revenue obtained from additional sales). For structured decisions it is possible and desirable to develop computer programs that collect and combine the data, thus giving the process a high degree of consistency. However, because these tend to be routine and predictable choices, a DSS is typically not needed for highly structured decisions. Instead, there are any numbers of automated tools that can make the decision based on the predefined criteria.

At the opposite side are *unstructured* decisions. These decisions have the same components as structured ones; however, there is little agreement on their nature. For instance, with these types of decisions, each decision maker may use different data and processes to reach a conclusion. In addition, because of the nature of the decision, there may also be few people that are even qualified to evaluate the decision. These types of decisions are generally the domain of experts in a given field. This is why firms hire consulting engineers to assist their decision-making activities in these areas. To support unstructured decisions requires an appreciation of individual approaches, and it may not be terribly beneficial to expend a great deal of effort to support them. Generally, unstructured decisions are not made regularly or are made in situations in which the environment is not well understood. New product decisions may fit into this category for either of these reasons. To support a decision like this requires a system that begins by focusing on the individual or team that will make the decision. These decision makers are usually entrusted with decisions that are unstructured because of their experience or expertise, and therefore it is their individual ability that is of value. One approach to support systems in this area is to construct a program that

simulates the process used by a particular individual. These have been called ***Expert Systems***. **It is probably not the case that an expert decision maker would be replaced by such a system, although it may offer support in terms of providing another perspective of the decision**. Another approach is to monitor and document the process that was used so that the decision maker(s) can readily review what has already been examined and concluded. An even more novel approach used to support these decisions is to provide environments that are specially designed to give these decision makers an atmosphere that is conducive to their particular tastes, a task well suited for a DSS. The key to support of unstructured decisions is to understand the role that individual experience or expertise plays in the decision and to allow for individual approaches.
In between are *semi-structured* decisions, most of decision support systems are focused on. Decisions of this type are characterized as having some agreement on the data, process, and/or evaluation to be used, but there is still a desire not to place too much structure on the decision and to let some human judgment be used. An initial step in analyzing which support system is required is to understand where the limitations of the decision maker may be manifested, i.e., will it be in the data acquisition portion, or in the process component, or possibly in the evaluation of outcomes. In any case, **DSS are designed to support but *not* replace human decision makers in semi- or unstructured situations**. They are aimed at assisting decision making by providing a flexible set of tools and capabilities for analyzing important blocks of data, by presenting information and interpretations for various alternative, and by suggesting a possible decision to make. At the end, there is still the need of human judgment for the final decision.
In practise, a DSS can assure several advantages. Some of such advantages are reporting in Table 3.I (Liping Sui, 2005).
These can be obtained by a decision support provided in several ways, such as:

- *User alert*, alerting the user to a decision-making opportunity or challenge;
- *Problem recognition*, recognizing problems that need to be solved as part of the decision making process;
- *Problem solving*, providing alternatives for the possible solution;
- *Facilitating/extending the user's ability to process knowledge*, acquiring, transforming, and exploring the knowledge;

Such functionalities should be supplied always in accordance to the requirements a DSS should fulfil. The necessary and relevant requirements can be individuated in relation to the stakeholders involved. The *decision-makers* or *users* are primarily concerned with what a DSS can do for them; their focus is on the problem-solving or decision-making tasks they face, hence they will assess the system in terms of the assistance they receive. In this perspective, important requirements are ease to use, robustness, quickness to response, completeness on important issues. On the other side there are the DSS *managers* and *builders*, who are involved in the design and configuration of the system. They are basically concerned with the problems of implementation, control and maintenance of the DSS; thus, for them, a DSS should be easy to control, adaptive, easy to integrate, and flexible.

3.1.1 Decision Support Models

Several classifications of DSS have been presented and discussed in the literature. An early classification was presented by Alter in 1980 (Alter, 1980): DSS were divided into seven categories depending upon the generic operations they perform, independently to the problem type, the functional area or the decision perspective. *File drawer systems*, *data analysis systems*, *analysis information systems*, *accounting and financial models*, *representational models*, *optimization models* and *suggestion models* are the categories considered by Alter (Figure 3.1).

Table 3.I. *Advantages assured by the use of a Decision Support System*

Advantages of DSS usage	Description
Fast calculating	A computer allows the decision maker to do a lot of calculations at high speed at low cost (the human resource cost of high-level management is usually high) and make timely decisions, which is very critical in many cases such as stock dealing, marketing strategy and so on, so forth.
Aid in processing and storing information	Human's intelligence is restricted by the ability of handling and storing information. Human being could not be able to memorize all information without mistakes at any time.
Cognition limit	A DSS can help people make a quick visit and process mass stored information. Besides, it can be useful in reducing the problems of coordination and communication among group of people sharing different knowledge of the problem.
Cutting down expense	A DSS can increase the productivity of the supporting staff, whose support is necessary for making decisions, and the increased productivity means lower cost. Moreover, it allow group of users to communicate across different places.
Information support	Computer technology allows the decision maker access correct, timely and latest information, which can be stored in different databases, even located in different places. Moreover, the information can consist in multimedia data. The computer system can query, access, and transmit the needed information quickly and economically.
Quality support	DSS can improve the decision quality. For instance, it can evaluate more alternate programs, and quickly make risk analysis and collect experts' opinions at high speed and low cost (maybe these experts are in different places). In addition, a lot of specialized knowledge can even be drawn by computer system. With the system, decision-makers can carry out complex simulations, check every possible situation and evaluate different effects quickly and economically.
Knowledge discovery and knowledge sharing	By building a data *warehouse* and performing *data mining*, data statistics and data analysis based on it, the implicit knowledge about the problems at hand can be found, and the access, update, improvement, application and feedback of knowledge can be exercised through an excellent *knowledge-base* management system designed based on DSS. The decision support system can realizes an efficient, safe and cooperated knowledge management and knowledge maintenance through the reasonable definition to the enterprise business process and organization structure as well as the employee relationship.

However, it has been widely recognized that such categories can be logically grouped into a simpler classification scheme: the first three have been called *data-oriented* or *-driven*; the second three as *model-oriented* or *–driven*; while the latter suggestion models has been called *Intelligent* or *knowledge-driven* DSS (IDSS).

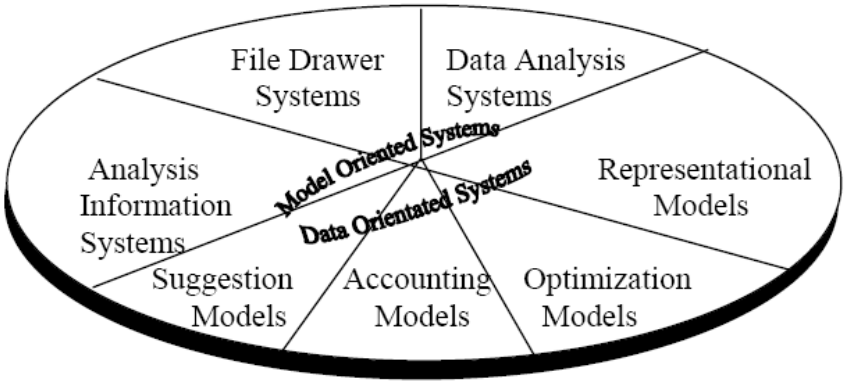


Figure 3.1. *Alter's Classification of DSS*

The number and structure of decision makers supported by the DSS have been used by Hackathorn and Keen (Hackathorn and Keen, 1981) to introduce the distinction in:

- ***Personal* or *Individual support***, this is exactly what it implies, decisions made by an individual acting alone, or with at the most, input and advice from others;
- *Group Support*, this refers to social units that are small (usually less than 20 people), established (i.e., have been working together long enough to have a sense of group identity, and most individuals in the group know each other at least in the context of the group's activities), and focussed on a particular function or line of work. ***Group DSS*** (GDSS) are principally focused on *communication*, of a kind ever required in individual decision making. Group members must either get together physically and talk, or exchange information using some medium in order to arrive at resolutions that can rightly be called "group" products. Group decisions also differ from individual decisions in the need for judgement rules to accommodate and resolve differences among the participants.
- *Organizational Support*, this refers to a collection of groups that are, together, engaged in a larger purpose than any single group encompasses or represents. The decision makers in organizational decision making under this definition are drawn from among the groups, but only a fraction of the membership of each group is included. Decision makers represent two social worlds: that of the group from which they are drawn, and that of the overall organization to which all the groups belong. ***Organizational DSS*** (ODSS) could be designed scaling up the capabilities of a GDSS, but taking care of the peculiar characteristics of organizational decision processes. These

usually address issues of larger social consequence and gravity than do group decision processes, are typically representative rather than inclusive of opinions of all group members, and are usually less socially "grounded" than group decision processes.

In 1999, Hättenschwiler (Hättenschwiler *et al.*) referred the classification to the relationship with the user, differentiating:

- ***passive** DSS*, a system that aids the process of decision making, but that cannot bring out explicit decision suggestions or solutions;
- ***active** DSS* which can bring out such decision suggestions or solutions;
- ***cooperative** DSS*, which allows the decision maker (or its advisor) to modify, complete, or refine the decision suggestions provided by the system, before sending them back to the system for validation. The system again improves, completes, and refines the suggestions of the decision maker and sends them back to her for validation. The whole process then starts again, until a consolidated solution is generated.

More recently, Power (Power, 2002) have used the mode of assistance as classification criterion for individuating the following eight types of DSS, which partially overlap the previous classifications:

- ***Data-driven DSS*:** These systems include file drawer and management reporting systems, data warehousing and analysis systems, *Executive Information Systems* (EIS) and *Spatial Decision Support Systems*; *Business Intelligence Systems* are also examples of Data-Driven DSS. Data-Driven DSS emphasize access to and manipulation of large databases of structured data and especially a time-series of internal company data and some times external data. Simple file systems accessed by query and retrieval tools provide the most elementary level of functionality. Data *Warehouse* systems that allow the manipulation of data by computerized tools tailored to a specific task and setting or by more general tools and operators provide additional functionality. Data-Driven DSS with *Online Analytical Processing* (OLAP) provide the highest level of functionality and decision support that is linked to analysis of large collections of historical data (Dhar and Stein, 1997). Professor Paul Gray argues that in approximately 1993, "the data warehouse and the EIS people found one another, with the data warehouses obtaining their needed application and the EIS people receiving a new breath of life from expanding beyond the pretty screen".
- ***Model-driven DSS*:** this second category includes systems that use accounting and financial models, representational models, and optimization models. Model-driven DSS emphasize access to and manipulation of a model. Simple statistical and analytical tools provide the most elementary level of functionality. Some OLAP systems that allow complex analysis of data may be classified as hybrid DSS systems providing modelling, data retrieval and data summarization functionality. Model-Driven DSS use data and parameters provided by decision-makers to aid them in analyzing a situation, but they are not usually data intensive. Very large databases are usually not needed for Model-driven DSS.

- ***Knowledge-driven DSS*:** The terminology for this third generic type of DSS is still evolving. Currently, the best term seems to be Knowledge-Driven DSS. Sometimes it seems equally appropriate to use Alter's original term, *Suggestion* DSS, or the narrower term *Management Expert System*. Adding the modifier "'driven" to the word knowledge maintains a parallelism in the framework and focuses on the dominant knowledge base component. Knowledge-driven DSS can suggest or recommend action to decision makers. These DSS are person-computer systems with specialized problem-solving *expertise*. As introduced in the previous section, the expertise consists of knowledge about a particular domain, understanding of problems within that domain and *skill* at solving some of these problems. A related concept is *Data Mining*. It refers to a class of analytical applications that search for hidden patterns in a database. Data mining is the process of sifting through large amounts of data to produce data content relationships. Tools used for building Knowledge-Driven DSS are sometimes called *Intelligent* Decision Support methods (Dhar and Stein, 1997). Data Mining tools can be used to create hybrid DSS that have major data and knowledge components.
- ***Communications-driven*** and ***Group DSS*:** Group Decision Support Systems (GDSS) came first, but now a broader category of Communications-driven DSS or groupware can be identified. This fifth type of generic Decision Support System includes communication, collaboration and decision support technologies that do not fit within those DSS types identified by Steven Alter. Therefore, Communications-driven DSS need to be identified as a specific category of DSS. We will call these systems Communications-driven DSS even though many people are more familiar with the term GDSS. A GDSS is a hybrid DSS that emphasizes both the use of communications and decision models. A Group Decision Support System is an interactive computer-based system intended to facilitate the solution of problems by decision-makers working together as a group. Groupware supports electronic communication, scheduling, document sharing, and other group productivity and decision support enhancing activities. We have a number of technologies and capabilities in this category in the framework – Group DSS, two-way interactive video, White Boards. Bulletin Board, and E-mail.
- ***Document-driven DSS***: a new type of DSS, a Document-driven DSS or *Knowledge Management System*, is evolving to help managers retrieve and manage unstructured documents and Web pages. A Document-driven DSS integrates a variety of storage and processing technologies to provide complete document retrieval and analysis. The Web provides access to large document databases including databases of hypertext documents, images, sounds and video. Examples of documents that would be accessed by a Document-Based DSS are policies and procedures, product specifications, catalogues, and corporate historical documents, including minutes of meetings, corporate records, and important correspondence. A search engine is a powerful decision-aiding tool associated with a Document-Driven DSS (Fedorowicz, 1993).

- ***Inter-Organizational*** or ***Intra-Organizational DSS***: a relatively new category of DSS made possible by new technologies and the rapid growth of the public Internet is an Inter-Organizational DSS. These DSS serve a company's stakeholders including Customers or suppliers. The public Internet is creating communication links for many types of inter-organizational systems, including DSS. An Inter-Organizational DSS provides stakeholders with access to a company's Intranet and authority or privileges to use specific DSS capabilities. Companies can make a Data-Driven DSS available to suppliers or a Model-driven DSS available to customers to design a product or choose a product. Most DSS are Intra-Organizational DSS that are designed for use by individuals in a company as "stand-alone DSS" or for use by a group of managers in a company as a Group or Enterprise-Wide DSS. The prefix "intra" means the DSS is used within a specific organization and "inter" means the DSS is used more widely.
- ***Function-specific*** or ***General Purpose DSS*****:** Many DSS are designed to support specific functions or types of businesses and industries. We can call such DSS function-specific or industry-specific DSS. A Function-specific DSS like a budgeting system may be purchased from a vendor or customized in-house using a more general-purpose development package. Vendor developed or "off-the-shelf" DSS support functional areas of a business like marketing or finance; some DSS products are designed to support decision tasks in a specific industry like a crew scheduling DSS for an airline. A task-specific DSS has an important purpose in solving a routine or recurring decision task. Function or task-specific DSS can be further classified and understood in terms of the dominant DSS component that is as a Model-driven, Data-driven or Suggestion DSS. A function or task-specific DSS holds and derives knowledge relevant for a decision about some function that an organization performs (e.g., a marketing function or a production function). This type of DSS is categorized by purpose; function-specific DSS help a person or group accomplish a specific decision task General-purpose DSS software helps support broad tasks like project management, decision analysis, or business planning. The most general purpose DSS are sometimes called DSS generators.
- ***Web-based DSS*****:** Finally, the deployment technology may be a mainframe computer, a client/server LAN or a Web-based architecture. All of the above generic types of DSS can be deployed using Web technologies and we can call these systems Web-based DSS. A Web-Based DSS is a computerized system that delivers decision support information or decision support tools to a manager or business analyst using a "thin-client" Web browser like Netscape Navigator or Internet Explorer. The computer server dial is hosting the DSS application is linked to the user's computer by a network with the TCP/IP protocol. In many companies, a Web-based DSS is synonymous with an intranet or Enterprise-Wide DSS. A company intranet is supporting a large group of managers using Web browsers in a networked environment. Managers increasingly have Web access to a data warehouse and analytical

tools. Also, Web technologies are the primary tools used to create Inter-Organizational DSS that support the decision making of customers and suppliers.

3.1.2 Decision Support System Structure

Although with different names and, sometimes, also different functionalities, three main components of a DSS can be identified as common to all the DSS structures proposed in literature. As shown in Figure 3.2, they are:

a) the *Database Management System* (DBMS): a DBMS serves as a data bank for the DSS. It stores large quantities of data that are relevant to the class of problems for which the DSS has been designed and provides logical data structures (as opposed to the physical data structures) with which the users interact. A DBMS separates the users from the physical aspects of the database structure and processing. It should also be capable of informing the user of the types of data that are available and how to gain access to them;
b) the *Model-Base Management System* (MBMS): the role of MBMS is analogous to that of a DBMS. Its primary function is providing independence between specific models that are used in a DSS from the applications that use them. The purpose of an MBMS is to transform data from the DBMS into information that is useful in decision making. Since many problems that the user of a DSS will cope with may be unstructured, the MBMS should also be capable of assisting the user in model building;
c) the *Dialog Generation and Management System* (DGMS) or *User Interface*: the main product of an interaction with a DSS is insight. As their users are often managers who are not computer-trained, DSS need to be equipped with intuitive and easy-to-use interfaces. These interfaces aid in model building, but also in interaction with the model, such as gaining insight and recommendations from it. The primary responsibility of a DGMS is to enhance the ability of the system user to utilize and benefit from the DSS. In the remainder of this article, we will use the broader term user interface rather than DGMS.

This is the first view introduced by Sprague and Carlson (Sprague and Carlson, 1982), which has been referred and detailed by many others authors. Power, for instance, used this list to explain how to build each of the eight DSS types discussed in the previous section in terms of the different emphases placed on the various components. He added *DSS architecture and network* component, referring it to how hardware is organized how software and data are distributed in the system, and how components of the system are integrated and connected. Data-Driven, Document-Driven and Knowledge-Driven DSS need specialized database component. A Model-Driven DSS may use a simple flat-file database with fewer then 1000 records, but the model component is very important. Experience and some empirical evidence indicate that design and implementation issues vary for Data-Driven, Document-Driven, Model-Driven and Knowledge-Driven DSS. Multi-participant systems like Group and Inter-Organizational DSS also create complex implementation issues.

For instance, when implementing a Data-Driven DSS a designer should be especially concerned about the user's interest in applying the DSS in unanticipated or novel situations. Despite the significant differences created by the specific task and scope of a DSS, all Decision Support Systems have similar technical components and share a common purpose, supporting decision making.
A Data-Driven DSS database is a collection of current and historical structured data from a number of sources that have been organized for easy access and analysis. We are expanding the data component to include unstructured documents in Document-Driven DSS and "knowledge" in the form of rules or frames in Knowledge-Driven DSS. Supporting management decision-making means that computerized tools are used to make sense of the structured data or documents in a database.
Mathematical and analytical models are the major component of a Model-Driven DSS. Each Model-Driven DSS has a specific set of purposes and hence different models are needed and used. Choosing appropriate models is a key design issue. Also, the software used for creating specific models needs to manage needed data and the user interface. In Model-Driven DSS the values of key variables or parameters are changed, often repeatedly, to reflect potential changes in supply, production, the economy, sales, the marketplace, costs, and/or other environmental and internal factors. Information from the models is then analyzed and evaluated by the decision-maker. Knowledge-Driven DSS use special models for processing rules or identifying relationships in data. A major issue today is whether DSS should be available using a Web browser on a company intranet and also available all the Global Internet. Networking is the key driver of Communications-Driven DSS.

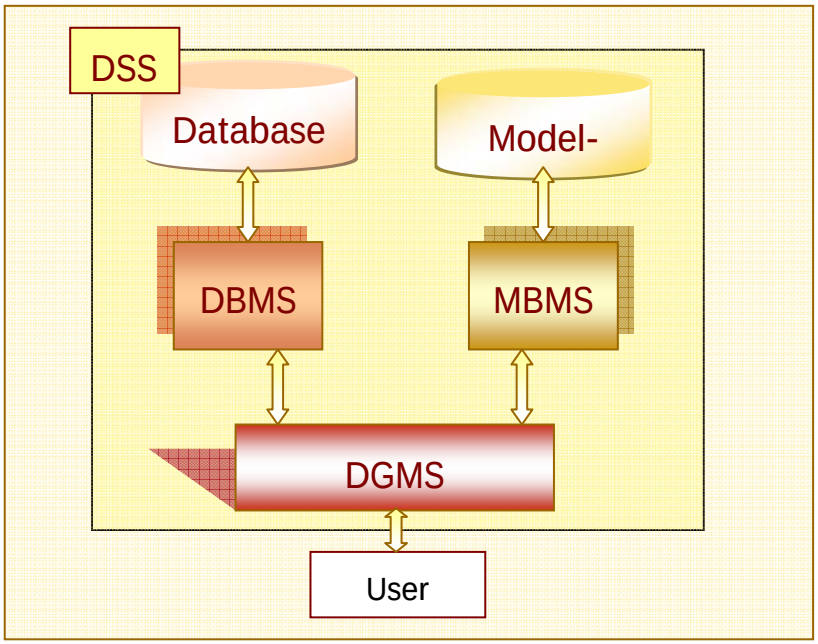


Figure 3.2. *The main DSS components: the database and the Database Management System (DBMS), the model-base and the Model-Base Management System (MBMS), the* ***Dialog Generation and Management System*** *(DGMS) or* ***User Interface.***

Marakas (1999) extended this view proposing a generalized architecture made of five distinct parts including a *knowledge base* (KB) component and the *user* himself. The latter choice is strictly related to the important role of the user as active actor of the system and its use. The KB is an essential component of Expert Systems, also called *Knowledge-based Systems* or *Intelligent Decision Support Systems* (IDSS) (Gadomski *et al.*, 2001). Actually, Sprague and Carlson considered the knowledge base as a part of the DBMS, describing the several models of data and, then, knowledge representation. However, data and knowledge are logically different for a number of reasons. Thus, we prefer herein Marakas' view, describing in more detail the five components. In particular, due to the importance of Expert Systems and KB modeling in the clinical applications, we will dedicate a major focus on the related issues.

3.1.2.1 The Database Management System

A database is an integrated collection of data, organized and stored in a manner that facilitates its easy retrieval. The structure of a database should correspond to the needs of the organization and should allow for access by multiple users and, when appropriate, for use by more than one application.

The sources of the data contained within the files may be quite diverse. The files may include data generated from internal transactions, external sources, and individual users of the DSS. Internal data normally come from an organization's transaction processing systems in the course of conducting the daily affairs of the organization. Depending on the problem context in which the DSS is operating, this data may be all that is necessary for the DSS to provide the necessary level of decision support. Often times, however, an internal data source is only one contributor of information to the DSS database.

External data sources are as broad as the imagination allows. They may include targeted collections of data for a particular industry, market research data. The good news for DSS designers and users is that most of the imaginable external data can be gotten with relative case and at little or no cost. The bad news is that there is so much of it.

Regardless of the source, the data contained within the files in the DSS database are organized into homogenous structures, or subunits, called records. In turn, the data contained within each record are organized into a series of data elements, or fields. The data element is the smallest unit of decomposition within the logical hierarchy and cannot normally be broken down further. Each record is constructed so that it contains a set of data elements related to a specific instance of the type of information contained within the file. Once these records are stored, they can be retrieved individually based on a unique piece of data, such as invoice number, or they can be retrieved as a unique subset, such as all records containing a particular product code. More important, they can be combined with other sources of organized data within the DSS database to allow for extensive analysis of relationships that might otherwise go unnoticed by the DSS user. This ability to combine data records from multiple sources gives the DSS a powerful, and

virtually infinite, repository of data from which the user can draw during the decision-making process.
One additional source of data that may be contained within the typical DSS database is referred to as individual-level or private data. Each instance of use of a DSS generates data about the decisions made, the questions asked, and the various combinations of data constructed for that particular problem context. In addition, individual heuristics, or rules of thumb, used by various DSS users may also be stored in the DSS database. These individual-level data elements and records often serve to create a unique personality for the DSS.
The multitude of data that can be organized into files and databases must be managed, and this important role falls to the database management system (DBMS). The DBMS has two main responsibilities:

- The coordination of all tasks related to storing and accessing information in the data-base and disseminating information to the community of DSS users;
- The maintenance of logical independence between the data contained in the DSS database and the DSS application.

Modern DBMSs possess a wide variety of capabilities and are generally managed by a skilled database administrator assigned specifically to that task. Recent DBMS developments include the facilitation of the process of integrating a large number of unrelated, or disparate, data sources into a single, accessible database known as a data warehouse. Data warehouses provide large amounts of data to the DSS in a form and manner that is more conducive to DSS use. The first responsibility of the DBMS – the coordination of tasks related to the storage, access, and dissemination of database information – involves several activities and functions. These include updating (in the form of adds, deletes, edits, or changes) data records and elements as transactions occur, facilitating the integration of data from various sources, and retrieving data from the database for queries or report generation. In addition, the DBMS performs many administrative functions related to DSS operation such as data security (control of unauthorized access, database error recovery, data integrity, and concurrency issues), the creation of personal or temporary databases for individual user experimentation and analysis, and the tracking of DSS usage and data acquisition, among others. Many of these activities are performed in the background and are intended to be invisible to the DSS user. Nonetheless, they are essential to the successful use of a DSS in a problem-solving context.
The second main responsibility of the DBMS, namely maintaining logical independence between the data and the DSS application, is also of extreme importance to the success of a DSS. The typical DSS scenario involves one or more users making decisions that are largely based upon information taken from a large pool of constantly evolving data. The DBMS must manage the physical organization and structuring of this data within the database and its associated files without constraining the logical, or apparent, arrangement of the data to either the DSS application or the DSS user(s). In addition, as new sources of data

are made available to the DSS, the DBMS must manage the integration of these disparate sources so that they appear to be all neatly organized in a common structure and location even when they are not. By maintaining independence between physical data structure and DSS applications, the DBMS allows for a much broader use of a single database, either by multiple DSS applications, multiple user groups, multiple problem contexts, or any combination thereof.

3.1.2.2 The Model Management System

A model is a simplification of some event or process constructed for the purpose of studying that event and thus developing a better understanding of it. The important characteristic of a model is that it is a simplified form of reality. It is intended to resemble the actual process or event as closely as possible but normally does not contain the same level or kind of detail as the phenomenon itself. In many cases, we can ascertain many characteristics of a process or event. Moreover, we can often predict the nature or outcome of that event under certain conditions without having to actually experience or recreate the event under study. The value of this capability can be found in the reduced cost, effort, and time derived from studying a model rather than the event itself.

The model base in a DSS is the modelling counterpart to the database. Just as the DSS database stores the data used by the DSS, the model base contains the various statistical, financial, mathematical, and other quantitative models the DSS uses to perform a variety of analyses. The model base is what differentiates a DSS from other computer-based information systems. The ability to run individual or combined models or to construct new models makes the DSS a powerful support toot in the problem-solving, environment.

The underlying models in a DSS can range in number, size, and complexity much like the data stored in a DSS database. To manage this contingent of analytical tools the DSS uses a model base management system (MBMS).

Two very important responsibilities of the MBMS are the execution and integration of the models available to the DSS and the modelling of user preferences. The user is normally required to provide data as input to a model or to provide certain parameters that may affect the model's execution. The MBMS takes care of this process by providing the user with easy access to the various models as well as facilitating entry of important parameters and data to the models.

Often the execution of one or more models requires the user to provide specialized syntax or commands to initiate the process. The MBMS generates the necessary command structure and locates the models for the DSS. The specific formatting requirements for data or parameter entry are also handled by the MBMS subsystem. Finally, the need to combine or integrate several individual models into a more complex one occurs regularly in DSS use. The MBMS facilitates this process of integration by providing the necessary links between the models as well as controlling the order of model sequencing and execution.

The modelling of user preferences is the process of collecting, organizing, and integrating the preferences, judgments, and intuitions of the individual DSS user into the modelling operation.

3.1.2.3 The Users' Interface

An *interface* is simply a component of a system that is specifically intended to allow the user to access the internal components of that system in a relatively easy fashion and without having to know specifically how everything is put together or how it works together easier it is for a user to access the system the better the interface. Also, the more common the interface, the less effort and training it takes for a user to move from one system to another.

The DSS interface is responsible for all interaction and communication with the user(s). Given this level of responsibility, the interface must not only include software components (such as menus and command languages) and hardware components (such as multiple monitoring or input facilities), but must also deal with factors relating to human interaction, accessibility, erase of use, user skill level, error capture and reporting, and issues relating to documentation, among many others. Because of this breadth and depth of responsibility, DSS experts regard the interface as the single most important component in the system. Without a good user interface, the power and functionality of the DSS remains forever inaccessible. The DSS interface has two components: the communication language and the presentation language.

The communication language or action language, component deals with activities associated with the user's direct dialog with the DSS. The various modes of data entry are included in this component. Data can be entered into the DSS by conventional methods such as keyboard, mouse, trackball, or touch pad. More recently, however, the communication component has been expanded to allow for a variety of not-so-common data entry mechanisms. Virtual reality devices such as biophysical input gloves, retinal scanners, and head position monitors are becoming common input methods. In addition, DSS applications using joysticks, voice recognition, and optical scanning devices are also being developed. Essentially, anything that can serve as potential input and can be captured as input is fair game for the communication language component of the interface.

Probably the most common aspect of the communication language is the menu. Menus provide the user with an organized and intuitive method of selecting, among,, the many functions, alternatives, commands, or outcomes available through the DSS. Properly organized and logically designed, menus can serve as guides for the inexperienced DSS user and as efficient vehicles of navigation for the DSS expert. Poorly designed menus, however, can render an otherwise powerful DSS unusable.

The presentation language component of the DSS interface is where all the action is. This is what the user actually sees, hears, or experiences during DSS use. Output devices such as printers, plotters, display monitors, audio monitors, and voice synthesizers are all part of this component. On-screen methods such as multiple windows, graphs, tables, charts, icons, messages, and audio triggers or alerts are also part of the presentation language. Just as the communication

language component serves as the basis for the user to transmit information and commands to the DSS, the presentation language component serves as the vehicle for the DSS to communicate with the user. These two components must be designed to work together in seamless harmony if the DSS is to be considered *user-friendly*.

3.1.2.4 The DSS User

No DSS can be considered functional or complete without the user. Unlike many other computer-based information systems, in the DSS the user is as much a part of the system as is the hardware or software. The user is commonly defined as the person, or persons, responsible for providing a solution to the problem at hand or for making a decision within the context the DSS was designed to support.

User Roles

In Alter's classification, the u*ser* is defined as the person who communicates directly with the DSS regardless of method or intention. The d*ecision maker i*s the person who ultimately makes the decision based, in whole or in part, upon the output from the DSS. An intermediary is a particular of user who serves as a filter or interpreter of the output from a DSS. The intermediary may work closely with the decision maker to assist in interpreting the DSS output during the various stages of the decision-making process. The maintainer, or operator is responsible or the daily operational aspects of the DSS such as keeping the system and its data up-to-date and in operational condition. Finally, the *feeder* provides data to the DSS but might not ever directly use the DSS as a decision support tool. One or several people or groups that regularly generate data relevant to the problem context the DSS was designed to support may fill the role of feeder. Alter suggests that the fewer people involved in the various user roles, the simpler the DSS become, to implement and use.

Patterns of DSS Use

The patterns of use described by Alter are beneficial in understanding how various users may interact with the DSS. In the *subscription mode* the decision maker typically receives *scheduled reports.* Such reporting is generally preformatted and generated on a regularly scheduled basis. The reports are often generated automatically the DSS and require no inquiry (other than the original one) on the part of the user. Moreover, these reports are usually generated through consolidation of existing information submitted from several sources throughout the organization and are received either through off-line methods such as hard copy or, increasingly, through asynchronous electronic methods such as e-mail or file transfer. An example of subscription mode usage would be the periodic receipt of a budget variance report.

In the *terminal mode*, the decision maker interacts directly with the DSS in an on-line manner. In this mode, the user determines and provides the input, directly manipulates the models in the model base, and directly receives and interprets the output.

A DSS can also be used in a *clerk mode*. Here, the decision maker is using the system directly but is not interacting with the DSS in an on-line, resume manner. Instead, the inputs, parameters, and requests are formed off-line and submitted via input coding forms or other electronic batch submission processes.
The fourth DSS usage category is the *intermediary mode*. Here the decision maker interacts with the DSS through one or more intermediary users. In this mode, the intermediaries are necessary due to the complex nature of either the analysis or parameter input process. In an extreme case, the DSS is positioned as a tool to be used by intermediary in the preparation of his or her final report to a decision maker. This mode of usage frees the decision maker from having either to interact with the "automated beast" or even know how the system works.

3.2 Intelligent Decision Support Systems: Expert Systems

As introduced at the beginning of the chapter, Expert Systems, also known as Knowledge-based Systems or Intelligent DSS, represent an important class of DSS. They are designed to face unstructured decision problems by simulating the process followed by a particular individual, the *expert*, who is usually entrusted with such decisions because of his/her experience or expertise (Chen, 2000).
The typical architecture of an EE is shown in Figure 3.3. The main components are: (i)a Knowledge Base, which contains a formal representation of the domain experts' know-how; (ii) an inference engine, able to infer new facts and suggesting decisions, using the knowledge in the KB and actual facts; (iii) the working memory, which maintains temporarily the actual facts; (iv) a database containing all the relevant information of the domain; (v) an explanation unit, which is responsible of explaining to the user why a particular decision has been suggested; (vi) the user interface, which is responsible of the system interaction with the user, and should be then carefully designed, as already discussed for DSS.
Two main design issues are involved in the definition of EE:

- Knowledge Representation
- Inference Engine Modelling.

Several methodologies developed for facing them are reviewed in the following.

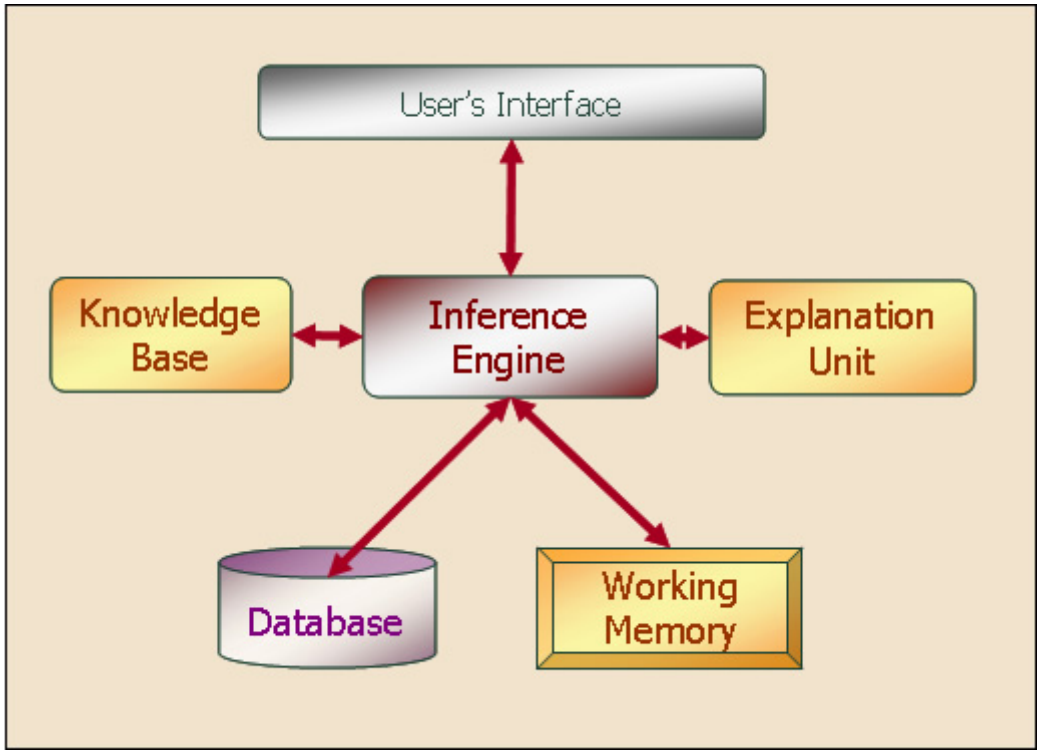


Figure 3.3. *The general architecture of an Expert System*

3.2.1 Knowledge Representation

The focal concern of an EE design is representation and management of knowledge, which has its own identity independently form the comprehensive functionality of the system. In order to better understand the importance of knowledge, it would be beneficial to take a look at the hierarchy of knowledge (Figure 3.4, revised from (Giarratano and Riley, 1998)). At the bottom of this hierarchy is the *data*, which is filtered from *noise*. Processed data is referred to as *information*, indicating or measuring how much we know from the underlying data. (In a loose sense, data can be viewed as the primitive form of information.)
Information used by agents to solve a problem will be referred to as *knowledge*. We say we can *access* information, but only an agent *possesses* the knowledge. Traditionally, knowledge has been associated with the concept of *belief*, which refers to statements that are inside the mind of an agent or can be inferred by the agent.

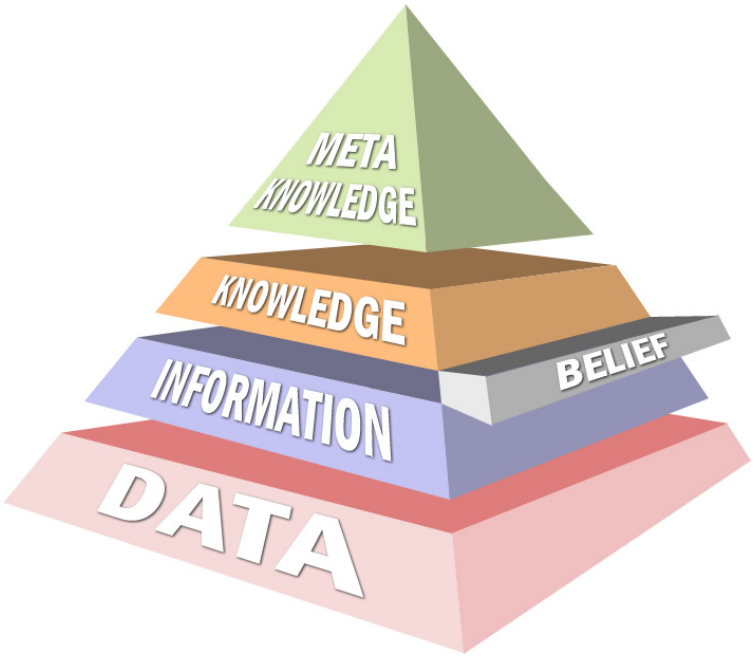


Figure 3.4. *The Knowledge Hierarchy*

These statements do not have to be true, and can be believed to varying degrees (Delgrande and Mylopoulos 1987; Poole *et al.*, 1998). Belief is thus concerned with the mental status of the agents. Without considering the mental status of the agents, it would be difficult to distinguish knowledge from information. Despite the differences between data and knowledge, however, both are useful in problem solving for decision support. In fact, a successful integration of management of both data and knowledge is the focal concern of EE development. Finally, at the top of the hierarchy, we have *meta-knowledge*. Just like knowledge can be used to manipulate the data, knowledge itself can be manipulated by meta-knowledge, which refers to knowledge about knowledge.

Knowledge could be ***procedural***, which refers to knowing how to do something, or ***declarative***, which refers to knowing that something is true or false. *Commonsense knowledge* refers to the knowledge a normal child possesses and has played an important role in computational intelligence. In addition, we can talk about ***tacit knowledge*** (or *unconscious knowledge*), which refers to knowledge cannot be expressed in language (for example, nodding). The study of knowledge is referred to as *epistemology*.

Knowledge Representation (KR) concerns understanding, designing, and implementing ways of formally coding the knowledge necessary for deriving other knowledge, planning future activities, solve problems that normally requires human expertise, and so on so forth.

The field of KR began, around 1958, with an investigation of how a computer might be able to represent and use the kind of simple common-sense knowledge. Since then, a lot of efforts have been dedicated to the task, resulting in the discipline known as *Information Engineering*.

Representing knowledge means to select a suitable *knowledge representation language* or *formalism* and define the *knowledge base*, which contains *statements* or *expressions* in the selected language for the problem at hand.
Fundamental problem of designing a knowledge representation language is the fundamental trade-off between

i. **expressivity:** the language is expressive enough to represent the important objects and relations in a problem domain;
ii. **tractability** (i.e., efficiency): the language allows reasoning about implicit information in a reasonable amount of time.

In the following, different KR formalisms are briefly described.

3.2.1.1 Rule-based Representation

A production system is a computer program with a form of artificial intelligence and it is generally composed of three elements:

- *Production rules*
- *Database*
- *Rules interpreter*

A production rule is divided into two parts, the left hand side (LHS) and the right hand side (RHS). LHS represents the premise (conditions), the RHS the consequent conclusion/action (deduction) when the preamble is verified, that is a modification of the data contained in the Database (adding or removing data).
An example of such a kind of rule is

IF *premise* **THEN** *conclusion/action*

The Database is a collection of data (facts) and it represents the state of the knowledge that will change on the base of the execution of the rules.
The interpreter has verify the premise (pattern matching process), by comparing them with the facts contained in the Database, and then, if the premise is satisfied, to approve the conclusion /execute the consequent action.

Supposing to have the rule:

IF *a person* "X" *has a parent* "Y" **AND** "Y" *has a parent* "Z"
THEN "X" *has a grandparent* "Z"

Rules are said "chained" when one or more conditions are shared.
Another module that helps in resolving possible conflicts among rules can be associated to the interpreter.

3.2.1.2 Frame-based Representation

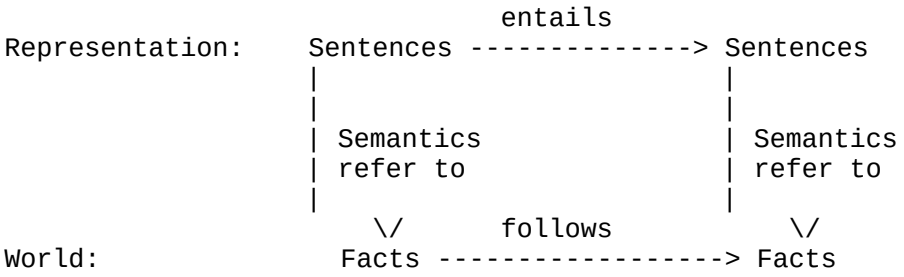
A frame is a data structure introduced by Marvin Minsky in the 1970s for knowledge representation (Minsky, 1975). Roughly similar to the object-oriented paradigm, they represent classes (called *frames*) with certain properties called *attributes* or *slots* whereas they do not have methods. Frames are thus a machine-usable formalization of concepts or schemata.
Like many other knowledge representation systems and languages, frames are an attempt to resemble the way human beings store knowledge. It seems our knowledge is stored in rather large chunks, and that different chunks are highly interconnected. In frame-based representations, knowledge describing a particular concept is organized as a frame. The frame usually contains a name and a set of slots.
The slots describe the frame with attribute-value pairs <*slotname, value*> or alternatively a triple containing <*framename, slotname, value*> in some order. In many frame systems the slots are complex structures that have facets describing the properties of the slot. The value of a slot may be a primitive such as a text string or an integer, or it may be another frame. Most systems allow multiple values for slots and some systems support procedural attachments. These attachments can be used to compute the slot value, or they can be triggers used to make consistency checking or updates of other slots. The triggers can be trigged by updates on slots.

3.2.1.3 Logic-based Representation

Logic is a formal system in which the formulas or sentences have true or false values (Sowa, 2000; Walker *et al.*, 1990).
Logic includes:

- **Syntax**: Specifies the symbols in the language and how they can be combined to form sentences. Hence facts about the world are represented as sentences in logic
- **Semantics**: Specifies what facts in the world a sentence refers to. Hence, also specifies how you assign a truth value to a sentence based on its meaning in the world. A **fact** is a claim about the world, and may be true or false.
- **Inference Procedure**: Mechanical method for computing (deriving) new (true) sentences from existing sentences

Facts are claims about the world that are True or False, whereas a **representation** is an expression (sentence) in some language that can be encoded in a computer program and stands for the objects and relations in the world.
We need to ensure that the representation is consistent with reality, so that the following figure holds:



A sentence is *True* if the state of affairs it describes is actually the case in the world. So, truth can only be assessed with respect to the semantics. Yet the computer does not know the semantics of the knowledge representation language, so we need some way of performing inferences to derive valid conclusions even when the computer does not know what the semantics (the interpretation) is
To build a logic-based representation:

- User defines a set of primitive symbols and the associated semantics
- Logic defines the ways of putting these symbols together so that the user can define legal sentences in the language that represent true facts in the world
- Logic defines ways of inferring new sentences from existing ones

Depending on the possibility and constraints each logic is characterised by a particular expressive capability that can be denoted using a literal standard notation:

- $\mathcal{AL}$: points out the attributes logic and introduces conjunction and disjunction operators and universal and existential quantifiers;
- $\mathcal{C}$: describes the possibility to use the negation operator;
- $\mathcal{S}$: extends the $\mathcal{ALC}$ DL with the possibility of the definition of transitive properties;
- $\mathcal{H}$: provides the possibility to define rules hierarchies;
- $\mathcal{O}$: asserts the presence of the enumerator operation;
- $\mathcal{I}$: allows inverse properties;
- $\mathcal{F}$, $\mathcal{N}$ e $\mathcal{Q}$ defines cardinality, respectively functional, simple and qualified (ordered by ascending expressivity);
- $\mathcal{D}$: allows the use of datatype properties, data values or data types.

Several types of logic are available for handling different type of *concepts* and obtaining different types of conclusions. Table 3.I summarised some of the existing logical languages, underlining the types of concepts and conclusions they manage.

Table 3.II. *Examples of logic language able to manage different types of concepts and conclusions*

Language	Concepts modelled	Conclusion
Propositional Logic	Facts	True/false
First Order Logic	Facts, objects, relations	True/false
Temporal logic	Facts, objects, relations, time	True/false
Fuzzy logic	Facts, objects, relations	Degree of membership

3.2.1.4 Fuzzy Logic

Fuzzy logic was introduced in 1965 (Zadeh, 1965) in order to treat vague and imprecise information expressed in the natural language. A decisional process is subjected to uncertainty due to the nature of human expression of a thought that is unlikely synthesizable into a clear-cut division between what is black and what is white, what is true and what is false.

In fuzzy logic, sets have the peculiarity of having vague (fuzzy) boundaries, not clear-cut, in which there isn't a very precise boundary between the elements belonging and not. Each element is characterised by a membership value that denotes in which measure it is included in the set.

The possibility to shade (fuzzy) the boundaries of objects' sets grouped under a same label, allows a high flexibility in order to manipulate and handle information, hampered in general by the dichotomous logic.

In the fuzzy logic, for example, an element belonging to a set is not compromised to belong to another set, even if for the classic logic those sets are mutually exclusive.

In this approach the inferencing rules have a true degree based on the premise and this is good in all those situations in which it is not possible to have the absolute certainty of a fact.

In Figure 3.5, we have three functions *cold*, *mean* and *warm* that map the temperature of a place in a certain month of the year. The value 17 °C is in the border between *cold* and *mean*, in particular it belongs to the class *cold* at 41% and to the *mean* one at 59%.

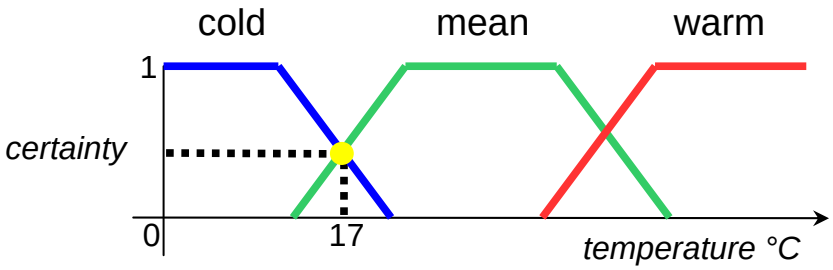


Figure 3.5. *Fuzzy membership functions*

With fuzzy logic we have a closer correspondence to a real situation, a more flexible approach versus a classic logic that in this case would be characterised by sharp jumps.
Fuzzy models are widely used because they allow catching the vagueness of phenomena: various fuzzy approaches for data analysis have been proposed (fuzzy clustering, correlation analysis, correspondence analysis, synthetic evaluation, adaptive techniques) and the research in this field is still active in order to improve the knowledge transfer to fuzzy systems.

3.2.1.5 Network-based Representation

The network-based representation relies on the use of *Semantic Networks*, formalism for representing knowledge as a graph, where nodes represent objects, concepts, and states, while arcs represent relationship among nodes (Woods, 1975; Brachman, 1979; Niemann *et al.*, 1990).
Computer implementations of semantic networks (see Figure 3.6 for an example) were first developed for artificial intelligence and machine translation, but earlier versions have long been used in philosophy, psychology, and linguistics. What is common to all semantic networks is a declarative graphic representation that can be used either to represent knowledge or to support automated systems for reasoning about knowledge. Some versions are highly informal, but other versions are formally defined systems of logic.
The most common kinds of semantic networks are the following:

1. *Definitional networks* emphasize the *subtype* or *is-a* relation between a concept type and a newly defined subtype. The resulting network, also called a *generalization* or *subsumption* hierarchy, supports the rule of *inheritance* for copying properties defined for a supertype to all of its subtypes. Since definitions are true by definition, the information in these networks is often assumed to be necessarily true.

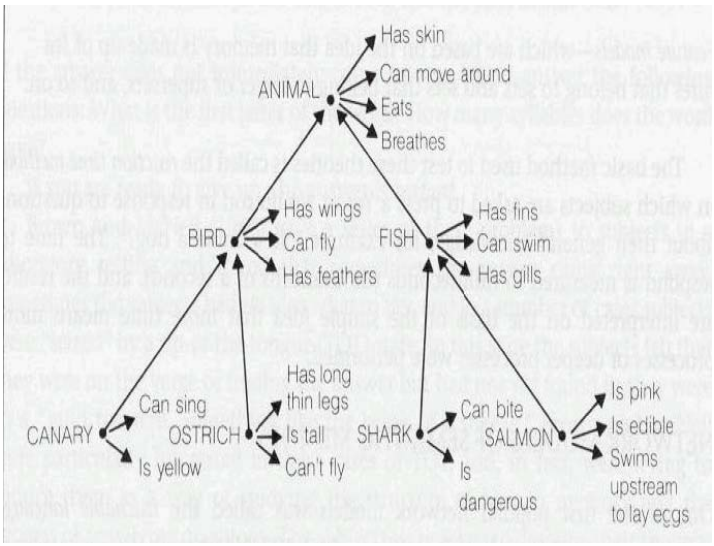


Figure 3.6. *An example of semantic network*

2. *Assertional networks* are designed to assert propositions. Unlike definitional networks, the information in an assertional network is assumed to be contingently true, unless it is explicitly marked with a modal operator. Some assertional netwoks have been proposed as models of the *conceptual structures* underlying natural language semantics.
3. *Implicational networks* use implication as the primary relation for connecting nodes. They may be used to represent patterns of beliefs, causality, or inferences.
4. *Executable networks* include some mechanism, such as marker passing or attached procedures, which can perform inferences, pass messages, or search for patterns and associations.
5. *Learning networks* build or extend their representations by acquiring knowledge from examples. The new knowledge may change the old network by adding and deleting nodes and arcs or by modifying numerical values, called *weights*, associated with the nodes and arcs.
6. *Hybrid networks* combine two or more of the previous techniques, either in a single network or in separate, but closely interacting networks.

Some of the networks have been explicitly designed to implement hypotheses about human cognitive mechanisms, while others have been designed primarily for computer efficiency. Sometimes, computational reasons may lead to the same conclusions as psychological evidence. The distinction between definitional and assertional networks, for example, has a close parallel to the distinction between *semantic memory* and *episodic memory*.

3.2.1.6 Workflow-based Representation

Workflows represent a useful and powerful instrument for knowledge representation when adopting a planning approach. A flowchart of tasks, states, events and actions are used to mainly formalize *procedural knowledge* (Peleg *et al.*, 2002).

Workflow problems can be modeled and analyzed using graph-based formalisms where structures are built on the standardized graphical notations (using nodes and directed arcs). Guidelines and protocols are often provided in a form of graphical flowcharts that present the sequence and flow of the medical tasks (Figure 3.7). The obvious advantage of the flowchart representation is that it is a format that is familiar to clinicians and at the same time easily interpreted by machines.

Workflow diagram systems are defined as “systems that help organizations to specify, execute, monitor, and coordinate the flow of work cases within a distributed office environment”. There are two logically separated parts of the workflow system:

- the workflow modeling component
- the workflow execution component (engine + interface)

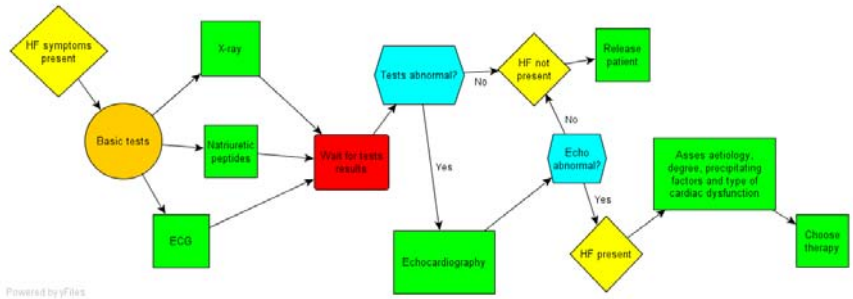


Figure 3.7. *A workflow diagram: different shape correspond to different functions.*

The workflow modeling component should provide a user friendly graphical interface for building flowcharts and manual editing of the flowchart parameters. The output from modeling tool is a structured knowledge written in one of the workflow formalizing languages. That knowledge is "executed" by the platform-embedded workflow execution component.

A workflow management system is a software component that given in input a formal description of business processes takes care of maintaining the state of the processes execution by delegating the specific activities to persons and applications.

The automation of the business process is obtained codifying the relationships among the procedures and mapping the sequences of actions needed in order to obtain a result on the base of the starting conditions.

Each process can be defined with the actions that have to be performed by the participating actors and with the evolution of the states in the flow.

The atomic part of the workflow is said "activity" of the process. Example of activities are taking a decision, sending a message to a person, printing a document.

Each process is described with the sequence of the set of activities performed.

The sequence of the activities can be linear but also ramified or net-ified, processes can also be executed in parallel, with different priorities or alternatives can exists.

Several contemporary instances can be performed following the same defined workflow: each running instance has its own life, that is it has its related information such as a start/end, a state, a history, etc...

Main categories of workflows applications are distinguished by functionality and capability:

- Production
- Autonomous Workflow Engines
- Embedded Workflow
- Administrative
- Collaborative
- Ad-Hoc

Other categorizations are based on the transport mechanism such as Message-based, Web-based and Suite-based.

3.2.1.7 Ontology-based Representation

In the last decades, ontologies emerged as knowledge representation formalism, logically equivalent to the frame-base schema (Davis *et al.*, 1993; Chandrasekaran *et al*, 1999).

The computer science view of ontology is somewhat narrower, where ontology is the working model of entities and interactions either generically or in some particular domain of knowledge or practice, such as molecular biology or bioinformatics. The following definition is given in:

'An ontology may take a variety of forms, but necessarily it will include a *vocabulary of terms*, and some *specification of their meaning*. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms'.

Gruber defines ontology as *`the specification of conceptualisations, used to help programs and humans share knowledge'*. The *conceptualisation* is the couching of knowledge about the world in terms of entities (things, the relationships they hold and the constraints between them). The *specification* is the representation of this conceptualisation in a concrete form. One step in this specification is the encoding of the conceptualisation in a knowledge representation language. The goal is to create an agreed-upon vocabulary and semantic structure for exchanging information about that domain.

The main components of an ontology are concepts, relations, instances and axioms. A *concept* represents a set or class of entities or `things' within a domain. Concepts fall into two kinds:

- *primitive concepts* are those which only have necessary conditions (in terms of their properties) for membership of the class. For example, a globular protein is a kind of protein with a hydrophobic core, so all globular proteins must have a hydrophobic core, but there could be other things that have a hydrophobic core that are not globular proteins.
- *defined concepts* are those whose description is both necessary and sufficient for a thing to be a member of the class. For example, Eukaryotic cells are kinds of cells that have a nucleus. Not only does every eukaryotic cell have a nucleus, every nucleus containing cell is eukaryotic.

Relations describe the interactions between concepts or a concept's properties. Relations also fall into two broad kinds:

- *Taxonomies* that organise concepts into sub- super-concept tree structures. The most common forms of these are
  - Specialisation relationships commonly known as the `is a kind of' relationship. For example, an Enzyme is a kind of Protein, which in turn is a kind of Macromolecule.
  - Partitive relationships describe concepts that are part of other concepts.

- *Associative* relationships that relate concepts across tree structures. Commonly found examples include the following:
  - Nominative relationships describe the names of concepts.
  - Locative relationships describe the location of one concept with respect to another.
  - Associative relationships that represent, for example, the functions, processes a concept has or is involved in, and other properties of the concept.
  - Many other types of relationships exist, such as *causative* relationships.

The relations, like concepts, can be organised into taxonomies. Relations also have properties that capture further knowledge about the relationships between concepts. These include, but are not restricted to:

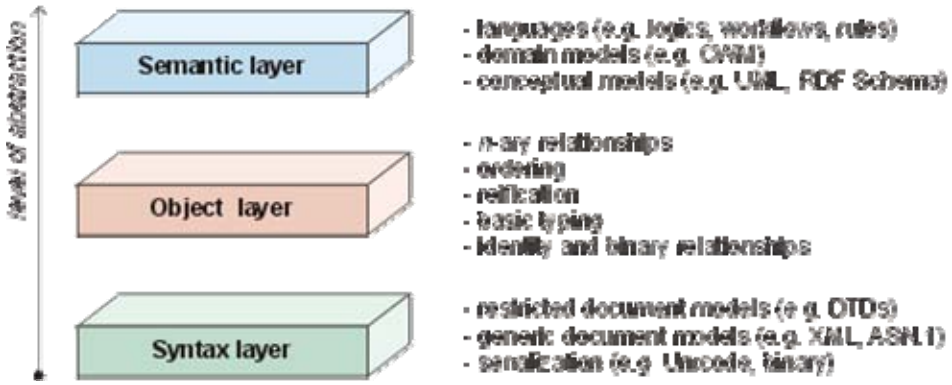
- whether it is universally necessary that a relationship must hold on a concept;
- whether a relationship can optionally hold on a concept;
- whether the concept a relationship links to is restricted to certain kinds of concepts;
- the cardinality of the relationship;
- whether the relationship is transitive.

The taxonomy relations always have this property.
Once this conceptualisation has been made concrete an ontology has been produced.
Instances are the `things' represented by a concept. Strictly speaking, an ontology should not contain any instances, because it is supposed to be a conceptualisation of the domain. Finally, *axioms* are used to constrain values for classes or instances. In this sense the properties of relations are kinds of axioms. Axioms also include more general rules.
Usually, the combination of an ontology with associated instances is considered as a knowledge base. However, deciding whether something is a concept of an instance is difficult, and often depends on the application. For example, Atom is a concept and `potassium' is an instance of that concept. It could be argued that Potassium is a concept representing the different instances of potassium and its isotopes etc. This is a well known and open question in knowledge management research.
Moreover, the original, main purpose of an ontology is to share common understanding of the structure of information between people and/or software agents, to reuse the domain knowledge and to make domain assumptions explicit.
For these reasons, ontologies are usually integrated with other KR formalisms, for instance rule- or logic-based.

- *A layered approach to Ontology management*

Creating a big encyclopaedia understandable to software agents without direct human interference might be quite interesting.
The advantage of such an approach is that it separates the basic (most reusable) knowledge and places this in an ontology layer, whereas a layer of models is positioned below.

The ontology layer is composed of a set of concepts representing abstract entities in the real world, relations among them based on external and function properties of the concepts.
This semantic layer, or knowledge representation layer, deals with conceptual modelling and knowledge engineering tasks.
The basic function of the object layer, or frame layer, is to provide applications with an object-oriented view of their domain.
The syntax layer is responsible for "dumbing down" object-oriented information into document instances and byte streams.



Each of the three layers has a number of sublayers. Every sublayer corresponds to a specific data modelling feature, e.g. aggregation or reification that can be logically implemented in many different ways. Plausible criteria for designing the layers and sublayers include grouping two (sub)layers if they have mutual dependencies, or merging a sublayer that has a single possible implementation option with an adjacent sublayer.

Syntax Layer

The main task of the syntax layer is to provide a way of serializing information content into a sequence of characters or bits. Any application that exchanges information with other applications or stores it persistently needs to structure the information carefully so that the recipient or reader can retrieve the information in its original form. Data structures used by applications are typically not just flat lists of uniform data elements. Therefore, additional markup mechanisms are required to preserve nested structure of data.

The syntax layer could be divided into three sublayers (bottom-up):

- *serialization*: data instances are serialized into byte streams. For example, XML documents are serialized as Unicode character strings, whereas ASN.1 uses a binary encoding.
- *generic document models*: applications structure information as nested data structures. XML provides a generic document model. Instances of this model can be manipulated using APIs like XML DOM.

- *restricted document models*: sometimes applications want to enforce structural constraints on the nested data structures they use. XML Document Type Definitions (DTDs) are an example of grammars for describing such structural constraints.

To reconstruct the objects and relationships from the representation used in the syntax layer, metadata is required. Obviously, two peer syntax layers need to agree on the metadata standard used for this purpose. For example, the emerging XML Schema standard can be deployed to extract objects and their relationships from an XML document. In this case, a specific XML schema would capture the information on how to identify objects. Another alternative is to choose a more generic or implied encoding standard.
On the other end of the communication link, the process is reversed and a high-level data structure is delivered to the application layer. Similarly to internetworking, every data modelling layer relies on a number of rules and conventions to exchange information with its peer, just as two corresponding networking layers deploy a specific communication protocol. For example, the syntax layer may require that the metadata be represented using the XML Schema standard. It is possible to call the list of such "protocols" (sets of conventions) a *model stack*. In this reference model, every data modeling language like UML, RDF etc. can be viewed as a specific model stack.

Object Layer

The purpose of the object layer is to offer applications an object-oriented view on the information that they operate upon. In contrast to data structures, objects have immutable object identity, i.e. change of an object's identity results in a different object. The very basic function of the object layer is to enable manipulation of objects and binary relationships between them. However, different applications may require more functionality of the object layer, depending on their complexity. We identified four additional sublayers that are often used. Here is a brief summary of the five sublayers:

- *identity and binary relationships*: every object-oriented model provides these features.
- *basic typing*: a simple abstraction mechanism. One object is used to "type" another object.
- *reification*: some information models (e.g. RDF, UML) require access to whole relations and individual links between objects.
- *ordering*: ordered relationships are integral part of some information models (e.g. UML).
- *n-ary relationships* are deployed in information models like SHOE.

Semantic Layer

Roughly speaking, the semantic layer provides interpretation of the object model used in the object layer. This objects, or surrogates, used in the object layer are

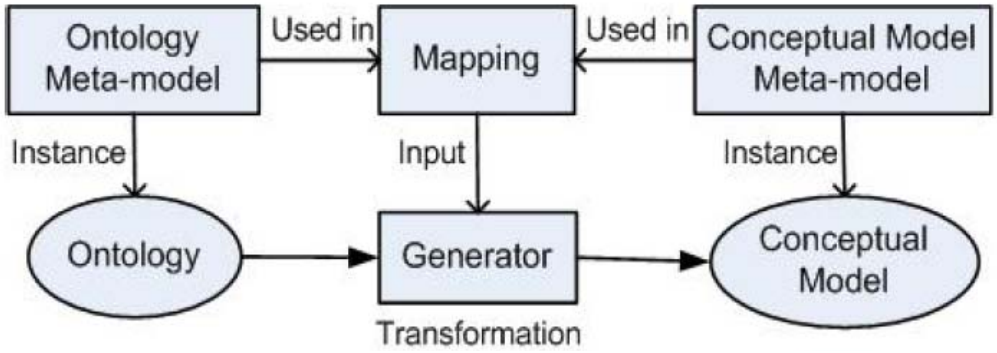
mapped onto physical and abstract objects like books, airplane tickets, database tables, logical formulae and paragraphs of text. The ultimate goal is to make applications interoperate in the semantic layer. The semantic layer is comprised of a number of rich and complex sublayers like:

- *conceptual models*: vocabularies for representing conceptual models (e.g. RDF Schema, UML Foundation/Core). Features of conceptual models may include elements like generalization, aggregation, cardinality constraints etc.
- *domain models*: deal with ontologies of a particular application domain, e.g. transportation, manufacturing, e-business, digital libraries, Web resources, etc.
- *languages*: instead of using a natural language, the machines on the Semantic Web convey information using formal languages. These languages can be highly specialized or serve a general purpose. Examples include workflow definition languages, Datalog, first-order logic, UML statecharts, SQL etc. Terms and expressions in these languages are first-class objects that can be manipulated on the object layer. In this way, applications can dynamically learn the semantics of previously unknown languages.

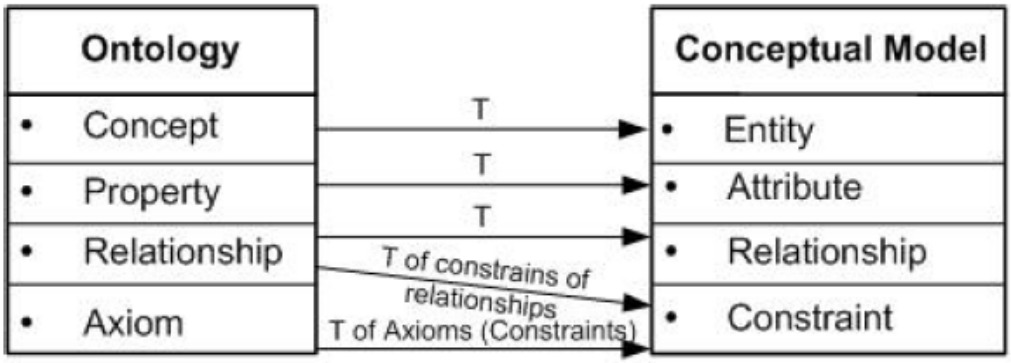
Common data interoperability in present applications is best achieved by using XML. XML (eXtensible Markup Language) is a meta-language used to define other languages. It describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them.
XML defines neither the tags nor grammar, which makes it completely extensible. It only requires that document must be well-formed in a tree structure, so it could be parsed by standard XML tools. In addition, the document can be structured to be valid. A valid document is one that conforms to its XML Schema, which defines grammar and tag set for specific XML formatting.
The Web Ontology Language (OWL) is a semantic markup language for publishing and sharing ontologies on the World Wide Web. OWL is developed as a vocabulary extension of RDF and is derived from the DAML+OIL Web ontology language.

Because conceptual models are intended to capture knowledge about a real-world domain, the meaning of modelling constructs should be sought in models of reality. Knowledge represented by ontology can be transformed into conceptual model. This transformation is possible, since concepts of the meta-model of ontology can be mapped into concepts of the meta-model of conceptual model.
Conceptual data models and ontologies are quite similar, as both consist of concepts, relationships between them and rules (in ontology – axioms), which are added in order to express other relationships between concepts and to constrain their intended interpretation. In conceptual modelling they are commonly called ‘constraints‘.

Ontology of a domain is richer then a conceptual model of the same domain but here there isn't any need to present some ontological aspects in a conceptual model.
For example, there is no possibility to present synonym and antonym relationships in conceptual model. But that knowledge about domain is necessary to understand it completely. Also it is important to note that several different conceptual models can be build from the same domain ontology. It depends on what we need to present in a conceptual model. Therefore, it is important to develop necessary rules for ontology transformation into conceptual model.
The ability to perform automatic model transformations implies that the mapping between the different concepts has to be developed only once for a pair of meta-models, not for each model instance.
To define the mapping between two conceptually different models requires a common basis that describes both the source and target domains of the transformation, and the transformation vocabulary. This common basis in this case is the meta-model.



Important is to develop necessary rules for ontology transformation into conceptual model.



1. Concepts that serve to designate a category or class of entities, events or relations are transformed into entities.
2. Properties that designate the characteristics of concepts are transformed into attributes of particular entities.
3. Relationships between concepts are transformed into relationships between entities.
4. Ontology axioms are transformed into constraints.

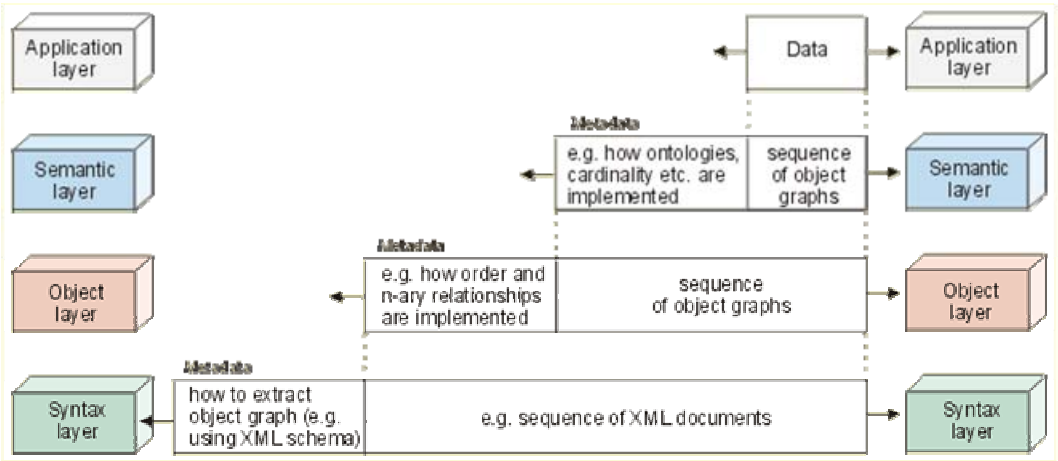
The ER diagram has been widely used as a means for describing conceptual or logical models.

There are the following type constraints in ER model meta-model:

1. Domain constraints are used to restrict the value space of data types. A value space is the set of values for a given data type. Some examples of constraints are: length, maxLength, minLength, e.g. Entity Constraints.

2. Entity constraints are used to restrict the instances of entities. All instances of an entity must satisfy its entity constraints. Entity constraints can be defined using allowed languages (English, OCL and XML). But nothing is written about possible constraint types.

In order to allow to a module of the HEARTFAID platform to obtain data for the execution of the specific demand, is necessary to supply the ability to demand to another application, organizing the data exchange software in a layered fashion, similarly to the approach taken in the internetworking.
Instead of forcing every application to deal directly with the details of semantics, structure and serialization of data, as in the networking architecture, the data is not exchanged directly between two peer layers.
The module of HEARTFAID platform is found over the application layer and uses the functions supplied from the below layers, every layer appends metadata needed to correctly interpret the data, and passes them to the layer below it.



Similarly to internetworking, such "protocols" as sets of conventions is a *model stack*. Every data modelling layer relies on a number of rules and conventions to exchange information with its peer, just as two corresponding networking layers deploy a specific communication protocol. For example, the syntax layer may require that the metadata be represented using the XML Schema standard. The semantic layer creates an object graph representing the entities and their relationships. It appends to the object graph the metadata needed to determine which ontology languages and ontologies are used, how they are implemented, how cardinality, aggregation etc. are expressed, and passes both the data and metadata to the object layer. The metadata appended at the object layer describes e.g. how ordered relationships or n-ary relationships are implemented or how typing of nodes is represented in the object graph. This information is forwarded to the syntax layer, which generates, for example, an XML document containing the object graph, and appends to it an XML schema needed to extract the object graph from the document.

3.2.2 Inference Engine Modelling

The inference engine is the system component devoted to handle knowledge for reasoning about available information, with the ultimate purpose of formulating new conclusions and answering questions.
Inference engine is structured in two levels, namely ***object level*** and ***meta level***. They can be considered as two individual systems, each with its own representation language, although they are strictly interconnected.
The object level is devoted to reasoning processes within the application domain; while the meta level is concerned with the *functional behaviour* of its object level, usually it makes use of different strategies for searching a solution, in order to reduce the computation efforts.
For better understanding the two-level structure and specifying the various types of inference engine systems, it turns out useful the concept of ***locus of action***, i.e. the place where the system is active at one point. We can, then, distinguish among three main classes:

i. systems with the locus of action mainly at the object level;
ii. systems with the locus of action moving from meta to object level alternatively.
iii. systems with the locus of action mainly at the meta level;

An inference engine has three main elements. They are:
1. an interpreter, which executes the chosen agenda items by applying the corresponding base rules.
2. a scheduler, the scheduler maintains control over the agenda by estimating the effects of applying inference rules in light of item priorities or other criteria on the agenda.

3. a consistency enforcer, The consistency enforcer attempts to maintain a consistent representation of the emerging solution.
The inference engine can be described as a form of finite state machine with a cycle consisting of three action states: *match rules*, *select rules*, and *execute rules*.
In the first state, match rules, the inference engine finds all of the rules that are satisfied by the current contents of the data store. When rules are in the typical *condition-action* form, this means testing the conditions against the working memory. The rule matchings that are found are all candidates for execution: they are collectively referred to as the *conflict set*. Note that the same rule may appear several times in the conflict set if it matches different subsets of data items. The pair of a rule and a subset of matching data items is called an *instantation* of the rule.
In many applications, where large volumes of data are concerned and/or when performance time considerations are critical, the computation of the conflict set is a non-trivial problem.
The inference engine then passes along the conflict set to the second state, select rules. In this state, the inference engine applies some selection strategy to determine which rules will actually be executed. The selection strategy can be hard-coded into the engine or may be specified as part of the model. In the larger context of AI, these selection strategies as often referred to as *heuristics*.
Finally the selected instantiations are passed over to the third state, execute rules. The inference engine executes or fires the selected rules, with the instantation's data items as parameters. Usually the actions in the left-hand side of a rule change the data store, but they may also trigger further processing outside of the inference engine (interacting with users through a graphical user interface or calling local or remote programs, for instance). Since the data store is usually updated by firing rules, a different set of rules will match during the next cycle after these actions are performed.
The inference engine then cycles back to the first state and is ready to start over again. This control mechanism is referred to as the *recognize-act cycle*. The inference engine stops either on a given number of cycles, controlled by the operator, or on a *quiescent* state of the data store when no rules match the data.

3.3 Clinical Decision Support Systems

In the clinical field, DSS are used to improved the quality of care and/or reduce cost without loss of quality. *Clinical Decision Support Systems* (CDSS) can be, then, defined as computerized applications that provide clinical knowledge and patient-related information, intelligently filtered or presented at appropriate times, to enhance patient's care, reducing costs and errors. Improved decision making can occur when the user is presented with reminders, critiques, facts and knowledge, or when an application allows the user to combine facts in a quantitative way that gives better results than one would normally make using known limitations of human reasoning.

There are several key words in the phrase ‘‘computer-based clinical decision support’’ and in the preceding definition. Here, the term computer is really shorthand for information and communication technologies, collectively. Clinical decision support can, of course, be provided by textbooks, teaching, manual feedback, and a variety of other methods; by computer-based, we mean that our focus is specifically on use of information and communication technology as the basis for providing it. By clinical decisions, we mean those that bear on the management of health and health care of an individual person (the patient). By support, we mean the aiding of rather than the making of decisions. By relevant knowledge we mean the selection of knowledge that is directly pertinent to the specific patient (Greens 2007).
CDSS has a number of characteristics that apply to most, if not all, of the many ways it can occur.

1. The general aim of CDSS can be one or both of the following:
    a. To make data about a patient easier to assess by, or more apparent to, a human.
    b. To foster optimal problem-solving, decision-making, and action by the human. The exact nature of a particular form of CDSS depends on its specific purpose.
2. The decision support is provided to a user—who may be a physician, a nurse, a laboratory technologist, a pharmacist, a patient, or other individual with a need for it. In some instances, the user may be a computer program rather than a human user. Many possible settings can give rise to this need, such as a problem arising in clinical practice, a health maintenance/preventive care question of a patient, or a training/ educational exercise.
3. A primary task of the computer is to select knowledge that is pertinent, and/or to process data to create the pertinent knowledge. To the extent that the computer can make the selection based on patient-specific data, the relevance of the CDSS to the individual patient is enhanced.
4. The selection of knowledge and processing of data involve carrying out some sort of inferencing process, algorithm, rule, or association method.
5. The result of CDSS is to perform some action, usually to make a recommendation.

Many attractive scenarios have been explored for use of CDSS over its approximately 45-year history. Actually, computers can be used for information retrieval, by providing search capabilities to find answers to specific clinical questions. They can do very basic error checks, enabling them to be guardians of safety—to detect problems when they occur or to prevent them altogether. A particularly valuable yet simple task is to perform data entry validation, as in the checking of a requested dose in a physician-entered medication order against predefined limits. Another practical and uncomplicated function is to continuously monitor new test results in a clinical laboratory, to identify conditions such as a critically low potassium level that require notification of the patient’s physician.

Yet another is to identify conditions that trigger reminders such as for scheduling an annual mammogram in a woman over 50 or for giving a flu shot to an elderly patient in the winter flu season. More complex uses are also of value. Among these, the idea of putting the computer to work to help make difficult diagnoses has been especially intriguing from the earliest days of computer use. In fact, from those earliest days up to the present, if one were to ask a layperson how a computer could be most useful for decision support in medicine, chances are that the person would say that it would be for making diagnoses. One of my own first exposures to CDSS in clinical medicine was the seminal paper in Science by Ledley and Lusted, published in 1959, entitled ‘‘Reasoning Foundations of Medical Diagnosis’’ (Ledley and Lusted 1959). This manuscript explored a combination of logical manipulation and probability, in particular, Bayes theorem, to identify most likely diagnoses given a particular set of findings. Over the ensuing four and a half decades, multiple applications and extensions of the approach have occurred, as well as development and evaluation of a number of alternative models for differential diagnosis (Caceres 1963; Pipberger et al. 1963; Warner et al. 1964; Lodwick 1965; Gorry and Barnett 1968; deDombal 1975). (It is interesting to note that among these activities, some of the earliest developments were in the area of electrocardiographic diagnosis (Caceres 1963; Pipberger et al. 1963), which involved signal processing and analysis of the ECG tracing, whereas the other efforts all dealt with clinical diagnosis requiring entry of findings by a user. As we will discuss further in subsection 1.4.2.2, the singularity of focus of ECG analysis and lack of need for human data entry likely contributed to the wider adoption and use of computer-based ECG interpretation today than other clinical diagnostic applications.)

Beyond diagnosis, the computer can support a variety of other complex decision-making tasks. It can help determine optimal workup strategy (Greenes *et al.* 1989) in sequencing of tests and procedures for evaluating a clinical problem (e.g., staging of colon cancer in an elderly man or evaluation of a breast lump in a young woman). It can assist in selecting treatment (Shortliffe et al., 1975), or in evaluating alternative treatment strategies (Kassirer *et al.* 1987) in order to select an optimal one for those conditions. It can be used to perform detailed treatment plans, in terms of dose calculations for chemotherapy (Knaup et al. 2002) or detailed 3D modeling and dosimetry calculations for radiation therapy (Ten Haken et al. 1995). It can provide estimates of prognosis and risk of complications for alternative treatments (Resnic et al. 2000; Inza *et al*. 2001).

In complex decision-making problem areas such as workup, diagnosis, treatment, and long-term management, just the ability to organize and coordinate the sequence of steps for performing various actions, evaluating results, and making choices of next steps is valuable. Thus, decision support in the form of clinical practice guidelines is of interest (Ohno-Machado et al. 1998; Shiffman et al. 1999; Miller et al. 2000; Peleg, et al. 2000; Greenes et al. 2001). Guidelines also can be used to embody best practices, with the hope that their use will improve health care quality, reduce variation, and improve efficiency and workflow.

In the following section, guidelines modelling methodologies are extensively review.

3.3.1 Guidelines Modelling

Several studies have shown the benefits of using clinical guidelines in the practice of medicine. Utilizing standard care plans, critical pathways and protocols in various clinical settings and tasks may lead to a reduction of practice variability and patient care costs, while improving patient care. Although the importance of these guidelines is widely recognized, health care organizations typically pay more attention on guideline development than guideline implementation for routine use in daily care, evidently hoping that clinicians will simply familiarize themselves with written guidelines and then apply them appropriately during the care of patients. However, some studies have shown that clinicians usually don't apply guidelines correctly during the care processes.
There is a clear need for effective guideline-based decision support tools as part of a fully integrated healthcare information system infrastructure, at the point of care and at the point of critiquing, which will relieve the current information overload on both care providers and administrators.
Guideline-based decision support systems aim to enable the latest clinical knowledge to be accessible and usable at the point of care and so make significant contributions to quality and safety in healthcare. These systems may improve the acceptance and application of guidelines in daily practice because the actions and observations of health care providers are monitored and advice is generated whenever they don't follow a guideline. Various studies, on different clinical settings and tasks, have shown that the use of these systems may significantly improves the quality of care, especially when used in combination with clinical information systems such as electronic patient record (EPR) systems.
Computer-based clinical guidelines are increasingly applied in diverse areas and a number of methods to support the computer-based guideline implementation have been or are being developed by the Health Informatics community as well as decision support systems that incorporate these guidelines. Guideline representation and formalization is a focus of researchers.

In order to define aspects regarding the process of developing guideline-based decision support system it's possible to identify 4 different steps:

- *Guideline modeling and representation.*
- *Guideline acquisition.*
- *Guideline verification and testing.*
- *Guideline execution.*

Guideline modelling and representation
The guidelines representation is a critical issue for the decision support systems development process. A formal and expressive model should provide an in-depth understanding of the clinical care processes addressed by guidelines, and thus will lead to (a) more rigorous methods of guideline development (for example, verification of a guideline's logical completeness and detection of ambiguity, inconsistency and redundancy), (b) more robust approaches for guideline implementation (for example, integration of guidelines with clinical workflow and

improvements in guideline maintenance), and (c) more effective techniques for guideline evaluation (for example, identification of variations in knowledge organization by different clinicians and resulting effects on their requirements for assistance during the process of decision making). A number of representation-related aspects can be formulated to fulfil the above-mentioned goals.
Primitives: the set of building blocks, used to represent the guidelines (e.g., rules, nodes, frames, etc.) must be expressive enough to capture the various aspects of a guideline.
Complexity: the representation must be able to represent various kinds of guidelines that may differ considerably in complexity and level of abstraction, for example by means of nesting or decomposition.
Knowledge types: guidelines contain a number of different knowledge types such as declarative knowledge (e.g., domain-specific knowledge) and procedural knowledge (e.g., inference or the method of decision support), which should be modelled separately.
Didactic and maintenance: as the content of a guideline is not static but may change over time, the representation must be able store didactic and maintenance information such as author names, versioning information, purposes and detailed explanations.
Language: the representation should be supported by a formal language (vocabulary, syntax and semantics), which has to be expressive enough to capture all the aspects, mentioned in the above points. In addition, a parser must be able to execute the guidelines in order to provide decision support, which requires a syntax that must meet execution-time requirements such as compactness and execution speed.
Local adaptation: it is important to make guidelines sufficiently general to be shared among different institutions. However, most guidelines undergo changes to make them acceptable to health care providers within a particular setting. These changes must be valid and consistent with the original guideline. When guidelines are updated on an (inter)national level, these changes have to be propagated to the guidelines on an institutional level while keeping the local adaptations intact, which requires sophisticated versioning and adaptation methods.

Guideline acquisition
The knowledge acquisition process is a very important issue in the development of guidelines and a number of knowledge acquisition tools (KA-Tools) have been or are being developed by many parties, used to acquire knowledge directly from a domain expert. These tools may facilitate the knowledge acquisition process by helping domain experts formulate and structure domain knowledge used in guidelines. As guidelines may change over time, also an update mechanism must be provided.

Guideline verification and testing
For acceptance of computer-interpretable guidelines in daily clinical practice, guidelines must be unambiguous and syntactically as well as semantically correct. For example, incorrect advice (e.g., false alarms) must be a minimum.

Verification tests, including the detection of various types of logical and procedural errors, may serve such a purpose. In addition, testing guidelines in a simulation environment (e.g., testing the guideline using a number of existing patient records) also increases their validity.

Guideline execution

To provide decision support, guidelines must be encoded in a format interpretable by automatic parsers that are incorporated in guideline execution engines. Guideline execution engines must be optimized to satisfy compactness and execution speed requirements. Furthermore, the architecture of the guideline execution engine must be system and application independent so that the guideline engine can be used in multiple clinical domains and in various modes (e.g., proactive versus reactive use).

A number of methods to support the guidelines computerisation process are based on guideline models which are capable of formalising medical knowledge as electronic applications that can be executed (or enacted) to generate patient-specific recommendations for clinical decisions and actions. Such methods employ different representation formalisms and computational techniques, for example, but some typical features are common for many representation models

Representation Primitives

Most of the actual guideline implementation models contain decisions and actions as their guideline representation key primitives. An *action* is a clinical task or intervention (for example a medication) that needs to be performed or is recommended during the guideline application process. A *decision* is a selection from a set of alternatives based on some criteria present in a guideline, for example, selection of a drug from a set of potentials. Many models also contain patient states or execution states as their primitives. A *patient state* is a description of a treated individual based on the actions performed and decisions made within the context of a guideline. An *execution state* is a description of stages of process with regard to the decisions and actions defined in a guideline. Most models support only one between explicit modelling of patient and execution states, at least as reported in the literature. In fact a strong relationship exists between patient state and execution state, as they are the two sides of the guideline application process. The patient state reflects the status at the patient side, while the execution state reflects the status at the system side. The compliances with the conditions for execution state transitions are guideline-specific and correspond to the patient states as defined above. Giving interventions induced by changes in guideline execution to a patient will change patient state. In this sense, patient state and execution state are closely related to one another, so most models with only one type of state represented are still rather expressive. However, as patient state can be affected by changes outside the control of a guideline application, patient state and execution state may diverge from one another. Different guideline implementation models use different ways in which actions, decisions, patient states, and execution states are represented.

Structure for Primitives: Temporal Constraints and Nesting

Primitives are just the elementary items in a guideline representation model and are not sufficient for complete guideline representation. Another important issue that needs to be addressed is structural arrangement of these primitives. Most of the actual representation models provide specifications of the temporal order that the primitive can be executed. Temporal constraints such as sequences, concurrences and iterations are used to encode guideline structures. Nesting of guidelines is another important representation feature that enables multiple levels of abstraction, which captures the composition of a complex guideline with subguidelines.

Patient Data Representation

The value of guidelines can be realized only through their application in clinical practice. Modelling of patient data is a critical issue for a guideline's integration with an EMR and an order-entry system, unless physicians read, memorize, and then use such guidelines on their own. One of the most important requirements for achieving such integration is a standard definition of patient data, which can then be mapped to implementation-specific database access methods through a standard interface. Several large projects have been devoted to building standard controlled medical terminologies, such as SNOMED and READ. How to incorporate these standards in a guideline representation so that they can be used to encode patient data thus becomes a critical issue.

Basically some kind of actions can de identified in the guideline application process. Decision-making is based on available patient data and other information. A data collection, that is a category of actions, needs to be performed before a decision can be made, although in many cases this is specified only implicitly by virtue of the use of data in a decision criterion. Patient state is often used as an entry or exit point for a guideline, but theoretically it may appear at any places in the guideline process flow. We thus define the patient state based on decisions made and actions performed in the context of a guideline. With this definition, the process to make a decision is in fact a confirmation of a patient state.
Finally interventions, another category of actions, are usually the cause of a change from one patient state to other patient states. The relationship among these representation primitives is shown in Figure 3-8.

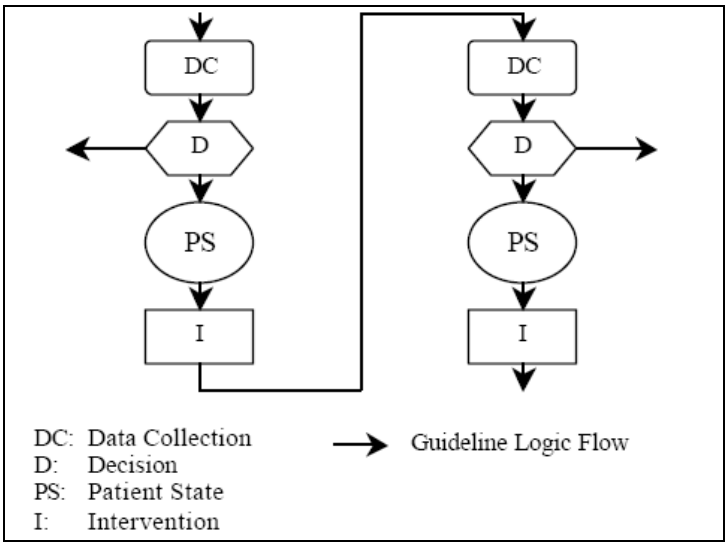


Figure 3-8. *Primitives, Modules and Guideline Logic Flow*

3.4 Design Issues

Moving from the general concept of DSS to the concrete design, development, maintenance, dissemination, and update of the system raises important issues and challenges.
What is really critical, in organizing the development of a DSS, is maintaining a user-needs focus, responding to those needs as they develop, change, and mature over time [S&C]. Fortunately, the very nature of DSS supports a development approach that tracks user needs closely. It is characterized by a small initial system which evolves and grows over time. The organizational questions this approach raises are:

- How can current needs susceptible to DSS be recognized?
- How can the likely extent of their growth be assessed?
- What type of DSS is required to support the needs, now and in future?
- Which are the minimum start-up capabilities required, both organizational and technical?
- What kind of plan can be developed to establish the long-term direction, yet respond to unanticipated developments in decision making needs and technical capabilities?

Once answered such questions, other relevant issues should be examined, namely the available tools for DSS developing, the technological facilities already available and used by the decision makers, and the evaluation of ad hoc or integrated approaches. Then, several tactical options can be adopted, including the following three:

- The *quick-hit*: if it is not clear that a general DSS capability is needed, but there is a recognised high-payoff area for decision support, develop a

specific DSS directly using the most appropriate tools, capture the benefits, then consider what to do next.

- The *staged development approach*: build one specific DSS, but with some advanced planning, so that part of the effort in developing the first system can be reused in developing a more general one.
- The *complete DSS*: develop, from the beginning, a full-service DSS and the structure to manage it.

DSS design requires a strong interaction among designers and decision makers to be supported. This is definitely true when designing an Expert System. The final goal is formalizing the knowledge, and the decision mechanisms and processes of decision makers. They should identify and define precisely their own experience and expertise, consisting of domain knowledge, heuristic rules, and reasoning mechanisms usually employed and simplifying expedients. The major effort of the designers should be spent in understanding such expertise, by means of a continuous interview process, and in finding the most suitable formalization.
Moreover, it is commonly recognized that a DSS would never be complete, since the knowledge and the model base are never complete, this means that it is always possible to improve them; moreover, the system can be prone to error, hence it should be necessary to add new knowledge and models to improve its performance.
These considerations highlight the necessity of following the *life cycle* of a DSS, especially when developing Clinical DSS (Greenes 2007). In particular, three main aspects of DSS appear to progress as if they have their own life cycles:

- *Knowledge generation, refinement, and update*
- *Knowledge management and dissemination*
- *Decision support method development and refinement*

These three life cycles are somewhat interdependent but they evolve at different paces, have their own constituencies, and involve separate processes.

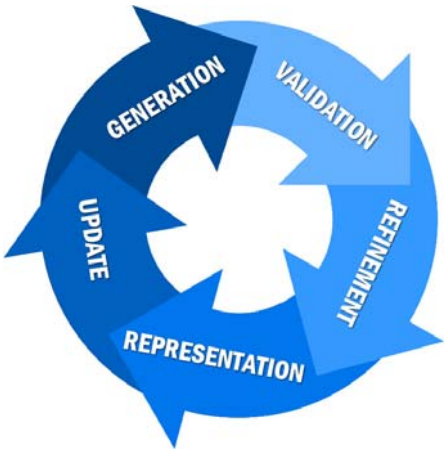


Figure 3-9. *The life cycle of knowledge generation, validation, refinement, and update*

Knowledge generation and validation. The knowledge underlying a DSS can be generated in a variety of possible ways. In general, the knowledge is initially unstructured and unassembled, or even only implicit, and must be extracted (from experts, from databases, or from the literature), organized and synthesized, analyzed for consistency and accuracy, and represented in an unambiguous form that can be computer-interpretable and acted upon. There may be gaps or overlaps with existing knowledge, calling for studies to refine the knowledge. Any synthesis of knowledge about a topic should have an appropriate expiration date, at which time the sources should be re-reviewed and the knowledge updated if necessary. Thus each item of knowledge must go through a continuous life cycle process (Figure 3-9).

Knowledge management and dissemination. Although we have considered the life cycle of individual knowledge content resources earlier, there is another task related to the corpus of knowledge in use and other knowledge that is being prepared for use. Imagine all the knowledge resources incorporated in or invoked by various applications throughout an enterprise. A subject expert in diabetes now wants to have the institution provide a set of checks and reminders for compliance with quality guidelines, such as periodic testing of a patient's HbA1c, eye examination, and foot examination. It is important not only to decide what the guiding knowledge should be, in terms of rules logic, order sets, and structured documentation templates, but also how this relates to similar knowledge resources that may already be implemented. What is needed is a means of curation of knowledge resources, to identify those existing items of knowledge pertaining to a topic of interest that are already in use, as an aid to the subject expert in creating new knowledge or refining existing knowledge, to avoid redundancy, to ensure consistency and avoid contradictions, and to recognize gaps where the opportunity for additional CDS may be needed. It may be useful to have a formal editorial process, with panels of experts, peer review and approval mechanisms in place, in order to accept new knowledge into a system. In short, a resource is needed to facilitate content management and collaborative authoring and review. Once knowledge is implemented in applications, it is necessary to keep track of where it is used, to be able to identify those instances when updates are required.

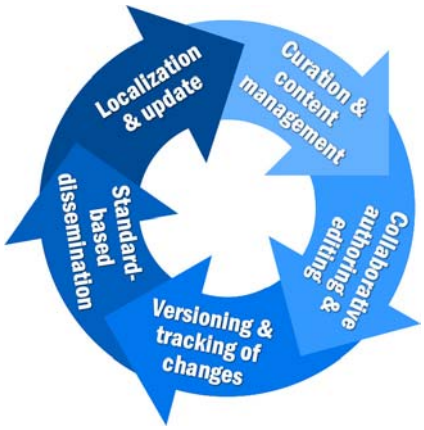


Figure 3-10. *The life cycle of knowledge content management and dissemination.*

.
On a broader scale, it may be desirable to maintain common repositories of computer-interpretable, unambiguous knowledge content (e.g., guidelines, or decision rules) for use across an enterprise that has multiple information system platforms. A still more ambitious goal would be to have regional, or national, or even international repositories of knowledge that are maintained and supported by government agencies, payers, or professional specialty or disease-focused organizations. For this to be useful, such knowledge resources need to be made available in a common format that is capable of being adapted to different platforms. This requires development and refinement of standards for representation of the knowledge. The notion of reuse would also benefit greatly from tools or standard approaches for adapting the content for different platforms, adapting to local customs or work processes, interfacing it to host patient databases, and invoking CDS through external services interfaces. Figure 3-10 depicts this life cycle process

Decision support implementation and evaluation. This process is a very complicated one. A life cycle is involved in developing the method for providing DSS (Figure 3-11), in terms of the decision model and intended decision support delivery approach. The decision model and its detailed methodology may evolve over time as nuances of the decision problem are recognized that require more or different parameters, or more complex computational or logical manipulations, or that require it to reformulate its advice for delivery in different ways to accommodate new needs. The optimal application environments in which the DSS is to be used must be determined, often by experimentation and pilot use, feedback, and refinement. The goals are to learn how best to integrate the DSS into specific settings, e.g. clinical, and to evaluate effectiveness. This process must continuously iterate. The representation scheme and mappings and modes of integration of DSS into host IT systems may evolve over time and must be updated. The methods for delivering DSS will depend on the availability and adoption of capabilities for integrating it into applications. For example, Computer based Provider Order Entry (CPOE) is most desirable as a platform for performing drug–drug, drug–lab, and drug–allergy checks to provide real-time feedback to physicians. Alerts and reminders require some kind of event or time trigger to cause the logic to be evaluated. Use of interactive data checks in structured data entry, groupings of knowledge into structured data entry forms and order sets, and methods for process and workflow optimization require appropriate application environments in which to provide their capabilities. All these methods of CDS need to be continually revised and updated as experience is gained with approaches that are successful versus those that are not.

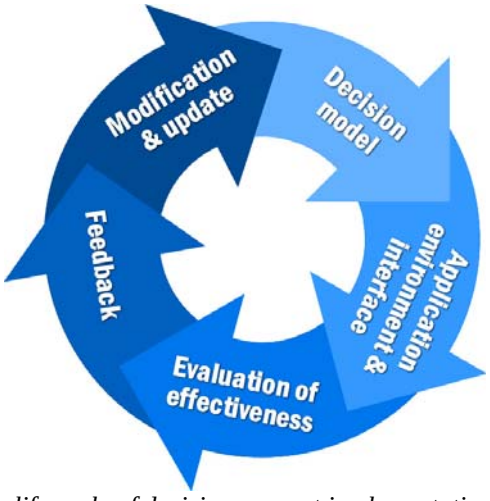


Figure 3-11. *The life cycle of decision support implementation and evaluation.*

Support for these three kinds of capabilities and their evolution through life cycle processes is a major part of the organizational, financial, and societal commitment that will be needed to make broad use of high-quality DSS a reality.

Dealing with the formalization stage, considerations can regard some specific issues, e.g. the knowledge representation language. In the previous section, we have discussed the most popular KR formalisms, specifying their main characteristics.

It is never easy to choose which knowledge representation technique (or combination) fits the needs of given system best. To make that decision it is very important to define problems that system will be used to resolve and it is very useful to know the possibilities and drawbacks of each technique of knowledge representation.

Representing the knowledge in one way may make the solution simple, while an unfortunate choice of representation may make the solution difficult or obscure.

Table II reports some advantages and disadvantages of different KR formalisms discussed in the previous section.

Table 3.III. *Advantages and Disadvantages of different Knowledge Representation Formalisms*

KR Language	Advantages	Disadvantages
Rule-based	▪ Decoupling rules from the programming code, useful when rules changes. ▪ ease of rules updating ▪ modularity and independence of the elements composing the system (rules, database, interpreter) ▪ ease of the rule language ▪ simplicity in exploring knowledge ▪ easy in constructing decision explanation	▪ Inefficient ▪ Less expressive
Logic-based	▪ with a semantics ▪ expressiveness	▪ Inefficient ▪ Undecidability Unable to express procedural knowledge ▪ Unable to do default reasoning ▪ No abduction
Frame-based	▪ domain knowledge model reflected directly; ▪ support of default reasoning; ▪ efficiency; ▪ support of procedural knowledge	▪ Lack of semantics ▪ Expressive limitations

3.4.1 Successful Decision Support Systems

In order to conclude the critic review of DSS and their development, it is worthwhile to analyze the factors make a DSS be successful. They can turn out useful in discussing and deciding the functionalities the HEARTFAID CDSS should supply.

Successful DSS and their subsystems must act intelligently and cooperatively in a complex domain with potentially high data rates and make judgments that model the very best human technicians.

It is also crucial that human technicians maintain control over the final judgments, either by focusing the system on particular reasoning goals, or by modifying the basic knowledge on which the system's judgments rely.

The high-level characteristics that a successful system should have are summarized in Table 3.I ().

Table 3.IV. *Success Factors of DSS*

Successful DSS High-level Characteristics	Brief Description
Interactivity	▪ Good interaction with other databases; ▪ Advanced interaction with the human users, allowing to explore the "space of possibilities" in a constraint-based way, instead of just providing the one "optimal" solution
Event and Change Detection	Recognition and effective communication of important changes and events.
Representation aiding	Information representation and communication in a compelling, informative, and human-centred way.
Error Detection and Recovery	▪ Control of typical reasoning errors made by people; ▪ Awareness of its own limitations and of situations for which it may not be as fully capable.
Information out of Data	Use of intelligent algorithmic techniques to massage data and generate information: ▪ extracting information from large amounts of data; ▪ providing the tools for handling small data sets, outliers, and other sources of error and confusion.
Predictive Capabilities	Prediction of actions effect on future performance: ▪ predicting the future environmental state; ▪ predicting the change in states caused by different decisions.

4. State of the Art of Technologies

4.1 Available Methods and Tools for Data Processing

4.1.1 Software packages for image processing and visualization

Below we describe the basic features of available commercial, free or open source software packages for image processing and visualization. We consider both packages with high prototyping value and packages mostly oriented towards development of end-user applications.

3D-Doctor
Commercial software for Windows platforms. Works with DICOM, as well as TIFF and BMP files (both 3D and 2D).It can be used as a viewer, but it also has 3D volume and surface rendering, restoration and more advanced image processing functions for CT, MRI, US, PET and other 3D imaging applications.
It is approved by FDA for medical imaging and 3D visualization applications.

3DVIEWNIX
It is non-free software, with complete source code available, developed by the University of Pennsylvania. 3DVIEWNIX has capabilities for visualizing, manipulating, and analyzing multidimensional, multimodal image information. It is designed to run on UNIX machines under X-windows. It uses a data protocol that is a multidimensional generalization of the ACR-NEMA standards. It provides useful tools for motion analysis of anatomical structures. Special filters are provided to allow for integration with VTK+ITK (see below).

ANALYZE
Non-free software by Mayo clinic. It provides an environment for the interactive visualisation and manipulation of 2-D, 3-D and 4-D biomedical images. An integrated set of tools is provided to allow data to be interrogated in both two and three dimensions. Three dimensional rendering tools are integrated with two dimensional orthogonal displays to allow real time reconstruction of conventional 2D. ANALYZE provides all the tools, including image registration, to truly support multi-modal image analysis. Tissue characterisation from multiple MRI, CT X-ray, and Nuclear medicine data is available as an interactive tool. Filtration program allow data preconditioning from statistical spatial filtering to minimise noise, and advanced 3D frequency domain deconvolution of the point spread function of a confocal microscopy system.

Interactive 2-D image display: Display of multiple images with variable size control; Mouse driven intensity windowing; Rapid generation of orthogonal images from 3-D volumes; Display of 3-D volume as a cube with control of size, intensity, range, angle-of-view and interactive dissections along orthogonal

planes; Generation and display of arbitrary oblique planar images through 3-D volumes; Interactive generation of "curved" images and/or radial image sections through images traced on orthogonal images; Rapid display of images in cine movie loops.

3-D Image segmentation: Semi-automatic segmentation using advanced morphology operations; Manual editing and automatic connection/deletion of multiple objects using region growing; 3D image editing and object definition; Multi-modal image classification and object definition.

Advanced 3-D image manipulation: Volume rendering using ray casting to display 3-D images from volumetric data. A complete suite of facilities is provided: Depth, depth gradient, grey scale and grey scale gradient shaded surfaces; Maximum intensity projection with optional depth weighting. Variable illumination and angle of view; Dynamic viewpoint manipulation; Transparency for overlying surface structures; 3D interaction between objects and orthogonal 2D slices; Multiple rendering parameters on different regions of the same display; Combined display of multiple segmented objects using different rendering parameters and colours.

Interactive surface labelling: Surface rendering for display of shaded surfaces from contours extracted from segmented image data; Surface smoothing and enhancement based on local neighbour characteristics within the data; Display and output of surface contour profiles; ASCII file output of surface normals for export to CAD/CAM or other design or prosthetic applications.

Multi-modal Image Analysis: Geometric image registration across multiple modalities using object surfaces or point files. Multi-modal image analysis and segmentation. Fused image generation and display. Cross modal object display - Bone from CT X-ray with soft tissue MRI.

Image & Data Manipulations: Linear combinations of images using algebraic operators; Pseudo transparent addition of multimodal data; Spatial and frequency domain image processing using standard and user defined filter functions. Histogram operations. Manual object segmentation using thresholding, tracing and erasing. Semi-automated, interactive boundary detection for object segmentation. Automatic edge contour extraction. 2-D and 3-D math morphology operators. 2-D and 3-D image transformation compression using wavelets.

Image measurement: Plotting of line and trace profiles including 3-D tracing. Region growing and spline region definitions. 2D and 3D region of interest definition. Selection and automatic sampling of regions of interest with image parameters. Interactive regional volume calculation. Regional shape and texture analysis. Data plotting and statistical analysis. 2-D and 3-D shape measurement tools. Multi spectral image classification tools for multimodal data characterisation.

Software development: Support of developments of Analyze program extensions simple user defined menu builder.

Data types and structures: As an inherently modality independent environment Analyze naturally allows the comparison and the fusion of data collected from different sites or scanners, or from different modalities.

AVS

Commercial package from Advanced Visual Systems. AVS is a visualization application software and development environment. AVS accepts data and attempts to create a visual display of the data in a variety of forms using different visualization techniques. AVS is structured around their concept of a module. A module is an independent computing element (C or FORTRAN) which is represented by a rectangular icon on the AVS screen. AVS comes with 110 modules, and the International AVS Centre provides access to a much larger set of modules contributed by the AVS user community. A range of data input, filter, mapper and data output modules are also included in AVS. Filters transform data into data, e.g. contrast stretch or edge detect. Mappers transform data into geometry, e.g. isosurface or arbitrary slice. And data output modules write data to files, send data to peripheral devices, or render data, e.g. displaying geometry, images and volumes on the screen.
AVS is currently available and used at CNR.

IDL

Interactive Data Language is a commercial package for the interactive reduction, analysis, and visualization of scientific data and images, suitable for cross-platform application development. IDL integrates a responsive array oriented language with numerous data analysis methods and an extensive variety of two and three dimensional displays into a powerful tool for researchers. IDL is useful in physics, astronomy, image and signal processing, mapping, medical imaging, statistics, and other technical disciplines requiring visualization of large amounts of data.

ImageJ

ImageJ is a public domain, Java-based image processing program developed at the National Institutes of Health. It offers some basic analysis tools.

ITK

ITK (the National Library of Medicine Insight Segmentation and Registration Toolkit) is an open-source software system to support the Visible Human Project. Currently under active development, ITK employs leading-edge segmentation and registration algorithms in two, three, and more dimensions.
The Insight Toolkit was developed by six principal organizations, three commercial (Kitware, GE Corporate R&D, and Insightful) and three academic (UNC Chapel Hill, University of Utah, and University of Pennsylvania).

It provides advanced methods (e.g. anisotropic diffusion filters and image segmentation filters using various formulation of levels sets methods), thus making it a comprehensive, powerful and up-to-date platform for image analysis. It has no visualization capabilities; however integration with VTK (see infra) is easily achievable.

The toolkit (written in C++) supports multiple language bindings, including such languages as Tcl, Python, and Java. These bindings are generated automatically using an auto-wrap process.

ITK is currently used at CNR.

4.1.2 Mathematical and Digital Image Processing

Mathematica by Wolfram offers a package called Digital Image Processing to bring the sophisticated analysis methods of Mathematica into image analysis. It includes several functions for color space transformations, connected component analysis, watershed transform, edge detection, smoothing, noise reduction, image morphology operations, iterative and non-iterative image restoration, spectral analysis. It may be suitable for development of new algorithm or to create prototyping procedures.

MATLAB: Image Processing Toolbox

MATLAB is a well-known high-level language and interactive environment to perform computationally intensive tasks in many research and engineering applications. It has powerful prototyping capabilities that significantly improve productivity. It is a commercial software available from MathWorks. A package, called Image Processing Toolbox (IPT), provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. Freely available bindings (MATITK) allow to access from the MATLAB environment some powerful algorithms of ITK, without dealing with low level programming issues.

MATLAB and IPT are currently available at CNR.

Mvox
General purpose commercial tool, developed by Anamedic for visualization and segmentation of a wide range of 2-4D medical images and graphics, for Linux and SGI platforms.
Mvox can handle images with different numbers of slices, colours and time steps. It has been used to visualize and segment: 2-3D medical images with stacks of slices (CT, MR, etc.), 2D remote sensing images with many channels and 2-3D colour images. In addition Mvox can also display and manipulate 3D surface

graphics (as used in CAD programs). 3D graphics can be exported in VRML format for use on WWW.

OSIRIS

Osiris is a viewer for Papyrus / DICOM images. It has been developed at the Digital Imaging Unit (UIN) of the Service for Medical Computing (SIM) of the Radiology department of the University Hospitals of Geneva (Switzerland).
Osiris is a multi-platform tool that exists for the PC/Windows (Windows 98 and up) and the Macintosh (Osiris for MacOS 9, OsiriX for MacOS X).
Osiris has basic tools such as contrast / intensity adjustment, ROI management, annotation tools, angle and distance measurements, area and volume computing, as well as more complex tools such as MPR reconstruction.
Osiris is well known worldwide and can be obtained free of charges.
The source code is available upon request.

VIDA

VIDA (Spanish for life or, more seriously, Volumetric Image Display and Analysis) is a comprehensive, non-free, package for the manipulation, display, and analysis of multidimensional image data sets, such as CT and MRI. VIDA is written in C, runs under the UNIX operating system, and uses the XView toolkit to conform to the Open Look graphical user interface specification. Available programs include: orthogonal sectioning, oblique sectioning, volume rendering, surface rendering, region of interest analysis.

Interestingly VIDA provides tool for cardiac mechanics analysis. In particular *Contour-Based Cardiac Mechanics* computes regional ejection fractions, regional wall thickness, wall thickening, etc. *Homogeneous Strain Analysis* was developed specifically to evaluate regional myocardial strain non-invasively through a magnetic imaging technique known as SPAMM by calculating the distortion of triangles generated from nodal points embedded within the myocardium.
Tube Geometry Analysis can be used for making 3-D geometric measurements, such as regional cross-sectional area, regional anterior-posterior length and lateral length of pre-segmented vessels or tubes.
Image Based Perfusion Analysis automates the analysis of cine X-ray CT images, thereby allowing one to rapidly compute physiologic data such as regional blood flow, regional tissue, blood and air contents, mean transit times, etc. Colour coded images of all physiologic parameters are generated and may be saved.

VisAD

VisAD is a Java component library for interactive and collaborative visualization and analysis of numerical data. The name VisAD is an acronym for "Visualization for Algorithm Development".

The system combines:

- The use of pure Java for platform independence and to support data sharing and real-time collaboration among geographically distributed users. Support for distributed computing is integrated at the lowest levels of the system using Java RMI distributed objects.
- A general mathematical data model that can be adapted to virtually any numerical data that supports data sharing among different users, different data sources and different scientific disciplines, and that provides transparent access to data independent of storage format and location (i.e., memory, disk or remote). The data model has been adapted to netCDF, HDF-5, FITS, HDF-EOS, McIDAS, Vis5D, GIF, JPEG, TIFF, QuickTime, ASCII and many other file formats.
- A general display model that supports interactive 3-D, data fusion, multiple data views, direct manipulation, collaboration, and virtual reality. The display model has been adapted to Java3D and Java2D and used in an ImmersaDesk virtual reality display.
- Data analysis and computation integrated with visualization to support computational steering and other complex interaction modes.

For these reasons, VisAd is suitable for use in a development environment. The code of VisAD is freely available.

VTK

The Visualization ToolKit (VTK) is an open source, free of charge, software system for 3D computer graphics, image processing, and visualization. VTK consists of a C++ class library, and several interpreted interface layers including Tcl/Tk, Java, and Python. It works very similarly to ITK. Filters are available to intertwine ITK/VTK filters pipelines. VTK supports a wide variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods. It also includes advanced modelling techniques such as implicit modelling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation.

It is possible to design new filters to be integrated in VTK, making the toolkit suitable for research and development environments.

VTK is currently used at CNR.

4.2 Available Instruments for Decision Support Services

4.2.1 Standards

What does open source software means?
The user of open source software has the following rights:
- to use it for any purpose
- to modify and personalize the source code,
- to redistribute copies of the program with the improvements guaranteeing the new source will be still open.

Some other advantages more than the direct ones already mentioned:
First of all, the license to use the software is for the most part free and when not it is anyway less expensive than the relative commercial one.
Results quality in general is higher, given the high number of qualified expert collaborating to the development, tests and documentation the complexity of the final product is surely major than a single commercial programming group could obtain.
Again because of is developed by hundreds and thousands of developers in general update is quick moreover the code is stable and sure.
Usage continuity will be assured by the developers' community while this cannot be the same for commercial software that depends on the state and choices of the producer in general based on continuing profitable and more diffused products
Open source software is often portable, that is it is allowable for different operative systems and in different languages and for different hardware.
Competition among different choices helps consumer at different levels: less risks of monopoly, low prices, more innovation and major specialization.
In general Open source software is more reliable than proprietary one: it is checked and verified by developers communities and being the code open a lot of users tests it with the purpose of solve eventual problems.
Considering Internet, it is open source the most spread software used to manage protocols, services, and applications such as Web (Apache HTTP), E-Mail (Sendmail), DNS (bind), secure communication (openssl), languages (Java), scripting languages (Perl, Python), operative systems (Linux) and so on: this is perhaps the most clear evidence of advantages open source offers.
Last but not least there are also a lot of ethical issues, see for example Stallman [Stallman]
Even using open source software privacy and security is guaranteed (see the cases of openssl and pgp).
A document about different kinds of open source licences can be found at the web site of the Open Source Organization [Open Source Organization].
A commercial product is a black box: it is difficult to check its security and besides it is neither customizable nor modifiable, it's difficult to redistribute.
The possibility to use a software in different environments from the original, named portability, it is an important requirement for data and programs.
Porting information can be a complex operation depending on the difference of the original and different environments: operative systems, programming

language and format of data are the key of success of distribution and sharing of the information among several and different users.
In the last decades the diffusion of the Internet network has fostered the spread of shared applications and information. In this scenario portability represents an important factor of success and its availability is often the most important choice. This means not only the possibility to use a program in different context but also the availability of standardized formats of data and metadata.
Data (and metadata) portability depends on its format: ASCII and UNICODE are examples of standard systems for codifying characters; XML W3C recommendation is a de facto standard for exchanging information. The adoption of data/metadata standards allows a wider number of persons and applications to use the same information.
Regarding programs the programming language is the main factor of portability: some programming languages are not portable (there is not an interpreter or a compiler able to translate the high level language into the one understandable by the computer exactly in the same way for all the systems). Examples of portable programming languages are C and Java.
Talking about data and metadata we have to focus our attention to standards.
Standards are important for several reasons:

- Broad industry agreement
- Interoperability
- Avoids vendor lock-in
- Open access
- Mandated
- Open, royalty-free standards

For our specific purposes a particular thought has to be done with Semantic Web Technologies.

From 1990 when the World Wide Web was created by Tim Berners-Lee and Robert Cailliau at CERN in Geneva, a huge amount of hypertext documents have been published on Internet.
Unfortunately most of this information is expressed in natural language and cannot automatically being understood and used by a software agent (a computer programme)
Describing Web resources with the use of metadata can help to modify this situation.
A metadata format usable directly by computer programmes can help to query, interpret and automatically elaborate data.
Semantic Web technology tries to provide a common way of facing this course/walk by formalising some standard specifications: among them Resource Description Framework (RDF) , RDF Schema, (RDFS), Web Ontology Language (OWL) and rules (SWRL, RIF) provides a way in order to formally describe concepts, terms and relationships of a problem domain.
Semantic Web technologies provide a benefit not only in the World Wide Web sphere but also in all the other domains that have to deal with data integration,

knowledge representation, and intelligent agents. The HEARTFAID project clearly fit into this sight.

What is Semantic Web?
It is intended as the possibility that information on the Web can be found, understood and used in order to achieve users' goals.
People is able to combine information got form the Web, considering for instance the goal of booking flight and a hotel he has to check several sites with different digital databases in order to find what he needs. To understand such heterogeneous information and to understand information contained in an image are trivial tasks for a machine. To provide examples another difficult task for a computer is to derive analogous information (John Smith is the same person as Smith John)

Talking about standards we cannot abstain from talking about W3C effort to recommend formats for the Web.
In the Figure 4.1 there is the W3C "Semantic Web Technology Stack", that is a vision about formats for the Semantic Web.

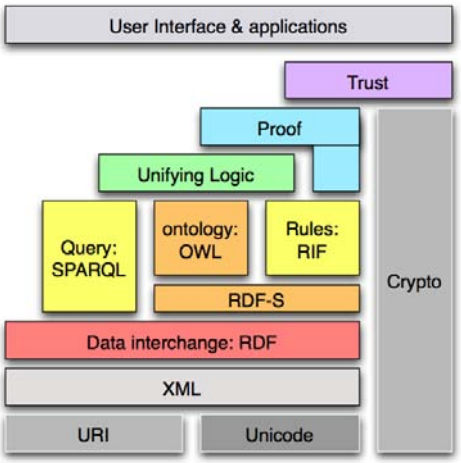


Figure 4.1. *The W3C Semantic Stack*

In order to render Internet information processable by a computer we need some globally recognized Web standards.
A URI, Uniform Resource Identifier, is used to univocally identify a resource in Internet.
The URI syntax specified in RFC 3986 includes as part of a URI a fragment identifier, separated by a number sign ('#'). The fragment identifier consists of additional reference information to be interpreted by the user agent after the retrieval action has been successfully completed.

XML, eXtensible Markup Language, is an extensible language realized in order to render easier the usage and the exchange of structured documents.
XML has been studied to allow and facilitate data exchange between different kinds of applications. The interest aroused by the new language is such that the most of software producers have adopted the XML format in theirs programmes.

RDF, Resource Description Framework is a language to semantically describe resources in Internet, expressing and linking data and metadata, to provide information about and relations between resources.
Resource Description Framework (RDF) provides a model to describe resources. It defines a resource as any object identifiable by a Uniform Resource Identifier (URI)
The RDF data model is represented as a set of statements or triples: a statement asserts a fact about a resource and it is specified by 3 parts: subject, predicate and object.
The subject is the resource described by the RDF expression, it can be a Web page, a set of Web pages, an anchor in a Web page but also any other object not accessible via Web but that can be referenced via a URI.
The predicate is the property, a characteristic, a specific aspect, characteristic, attribute, or relation used to describe a resource. Each property has a specific meaning, defines its permitted values, the types of resources it can describe, and its relationship with other properties. Properties associated to resources are identified by a name and take some values.
The object can itself be another resource or a literal (a string of characters or another primitive data type defined by XML).

RDFS, RDF Schema is a framework that provides the means of specifying vocabularies for RDF; it provides mechanisms for describing groups of related resources and their relationships.

Ontologies, in the field of artificial intelligence, are used to describe information and concepts from a particular domain and to derive relationships among these terms. OWL, or Web Ontology Language is the actual de-facto standard to define ontologies.

SPARQL, recursive acronym that stands for SPARQL Protocol and RDF Query Language, is a query language for getting information from such RDF graphs. It provides facilities to extract information in the form of URIs, blank nodes and literals, extract RDF subgraphs, and construct new RDF graphs based on information in the queried graphs.

RIF, Rule Interchange Format, specifies a format that allows rules to be translated between different rule languages and thus transferred between rule systems.

A significant rule language, SWRL (A Semantic Web Rule Language Combining OWL and RuleML), is a submission to W3C: it proposes a language that combines the expressivity of RuleML and OWL
This proposal extends the set of OWL axioms to include Horn-like rules. It thus enables Horn-like rules to be combined with an OWL knowledge base.
In logic, a Horn clause is a clause (a disjunction of literals) with at most one positive literal
RuleML covers the entire rule spectrum, from derivation rules to transformation rules to reaction rules. RuleML can thus specify queries and inferences in Web ontologies, mappings between Web ontologies, and dynamic Web behaviours of workflows, services, and agents.

Logical reasoning is needed to check the consistency and correctness of information and to infer conclusions that are not declared by initial facts.
Proof is needed to explain the steps of logical reasoning of the inference engine to infer new facts.
Trust is required by computer agents and services in order to authenticate and identify the reliability (trustworthiness) of information.
Considering the evolution of the technologies we can focus our attention on the central layer of the picture: queries, ontologies and logic.
Ontology is the science of "being" [Aristotle, Metaphysics], that tries to study what are the characteristic of being.
In information science, ontology describes a formal specification of a simplified view of the domain we want to represent. It is constituted by a vocabulary and shared meaning of it.
A system based on ontologies can allow data exchanging among programs, sharing information through a simple unification or translation of different representations, to use knowledge-based services, to facilitate exchange of information of people and machines.
Ontologies are used to classify resources, to identify them, and to define properties and relationships among them.
OWL provides three sublanguages:

- *OWL Lite* supports those users primarily needing a classification hierarchy and simple constraints. Tools supporting OWL Lite are simple to provide because of its narrow level of expressiveness: OWL Lite provides a quick migration path for thesauri and other taxonomies.
- *OWL DL* supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL.

- *OWL Full* is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

Considering this notation, the OWL language can be considered equivalent to a $\mathcal{SHOIN}(\mathcal{D})$ descriptive logic.

Logic, intended as formal logic, can be useful to Semantic Web in many scenarios: rule application and evaluation, facts inference in order to deduct/induct new facts, giving an explanation of a reasoning providing the trace of the step executed, to represent knowledge, to find contradictions inside the knowledge-base model, to query and combine heterogeneous information.

4.2.2 Process Management

This section is a list of open source workflow engines.

Twister is a whole open source workflow (or business process management) solution, written in Java, using the WS-BPEL standard. It is web services oriented but also supports other ways of interaction.

jBpm is a flexible, extensible workflow management system. Business processes, expressed in a simple and powerful language and packaged in process archives, serve as input for the jBpm runtime server. jBpm bridges the gap between managers and developers by giving them a common language: the jBpm Process definition language (jPdl).

Enydra Shark is an extendable workflow engine framework including a standard implementation completely based on WfMC specifications using XPDL (without any proprietary extensions!) as its native workflow process definition format and the WfMC "ToolAgents" API for server side execution of system activities

OpenSymphonyWorkflow can be considered a "low level" workflow implementation. Situations like "loops" and "conditions" that might be represented by a graphical icon in other workflow systems must be "coded" in OSWorkflow.

con:cern is a workflow engine based on an extended case handling approach. A process is described as a set of activities with pre- and postconditions. An activity is executed when its preconditions are met. It manipulates the process item, thereby creating postconditions. The process flow is determined at run-time.

Werkflow is a flexible, extensible process- and state-based workflow engine. It aims to satisfy a myriad of possible workflow scenarios, from enterprise-scale business processes to small-scale user-interaction processes. Using a pluggable and layered architecture, workflows with varying semantics can easily be accommodated. Processes can revolve around documents, objects or any other entity. The core werkflow engine can be accessed through a Java API, EJB, JMS, SOAP and other conduits.

ObjectWeb Bonita is a flexible cooperative workflow system, compliant to WfMC specifications, based on the workflow model proposed by the ECOO Team, which incorporates the anticipation of activities as a more flexible mechanism of workflow execution. Bonita is Open Source and is downloadable in LGPL License.

Bossa is a workflow engine written in Java. The engine is very fast and lightweight, uses a very expressive Petri net notation to define workflows, does not require a RDBMS and is very simple to use and to integrate with java applications.

The **Open Business Engine** is an open source workflow engine written in Java. OBE workflow definitions are written in XPDL, the WfMC's XML process definition language and are typically executed inside of a J2EE container.

The **Open for Business Workflow Engine** is based on the WfMC and OMG spec. It is a member of the Services Framework and is tightly integrated with the Entity Engine.
All changes to a process or activity are persisted real-time. Therefore, the engine does not run in a thread, it is simply a group of APIs and common objects which handle the flow. When a change to the workflow is made, the engine then processes that change. When finished, the engine returns. Hence, if the application does crash, or the system reboots, the workflow will continue right where it left off upon restart.

OpenWFE is an open source java workflow engine. It is a complete Business Process Management suite, with 4 components: an engine, a worklist, a webclient and a reactor (host for automatic agents). A python access library is available: your python application / client can interact with an OpenWFE REST worklist.

WfMOpen is a J2EE based implementation of a workflow facility workflow engine) as proposed by the Workflow Management Coalition (WfMC) and the Object Management Group (OMG).

XFlow is a pure J2EE platform for building, executing and managing business processes and workflows. It is a basis for building collaborative applications as well as integrating processes across an enterprise. XFlow has a small footprint but

is extremely powerful. It is designed to be easy to use from the development, deployment and management standpoints.

JFolder (formerly PowerFolder) is workflow server and development studio. It can be configured to work on J2EE application servers and a variety of persistence stores (databases).

The **Taverna** project aims to provide a language and software tools to facilitate easy use of workflow and distributed compute technology within the eScience community. As a component of the EPSRC funded myGrid project, Taverna is available freely under the terms of the LGPL.

Freefluo is a workflow orchestration tool for web services initially developed by IT Innovation but now available to all from the Freefluo Sourceforge Site. It can handle WSDL based web service invocation. It supports two XML workflow languages, one based on IBM's WSFL and another named XScufl that is under development as part of the Taverna Sourceforge project.

The **micro-workflow** framework targets developers who want to separate the control and logic aspects in their programs, thus making them flow independent. A well-factored flow independent application facilitates change because the most frequent business changes translate into process changes, thus leaving the code intact.

JFlower is a very light workflow handler. You can write complex flow, with conditions and other flow invocations, and JFlower evaluate each step of the flow to the end of the flow.

YAWL (Yet Another Workflow Language), an open source workflow language/management system, is based on a rigorous analysis of existing workflow management systems and workflow languages. Unlike traditional systems it provides direct support for all of the workflow patterns (http://www.workflowpatterns.com). YAWL supports the control-flow perspective, the data perspective, and is able to interact with web services declared in WSDL. It is based on a distributed, web-friendly infrastructure.

Syrup. An adaptive workflow engine based on simple concepts. It offers full persistence, so that work is never lost, and deals with partial failure in a distributed setup. Syrup is also used reliably as a distributed scheduler to replace cron.

PXE (short for Process eXecution Engine and pronounced like "pixie") is a runtime component for executing processes defined by the BPEL4WS 1.1 specification.

ActiveBPEL engine is a robust runtime environment that is capable of executing process definitions created to the Business Process Execution Language for Web Services (BPEL4WS, or just BPEL) 1.1 specifications.

AntFlow is a tool for the automation and scheduling of data system tasks, including those with complex dependencies and workflow logic. Antflow represents a new approach to simplifying system automation that leverages pipelines of hot folders chained together to perform a given task. Using XML, Antflow associates an automated task, such as data transfer, compression, or encryption, with a directory on the local system. Whenever a file is copied or written into the hot folder, the associated task is executed and the file is moved to the next hot folder in the pipeline for further processing.

Dalma. The heart of the engine is an ability to capture the execution state of a thread and resume it later. Many applications of today need to have a part of the program that waits for other entities. . Often there are multiple conversations running concurrently. Those are what we call "workflow" applications. Today, those applications can be written, but one can't write it very productively. Dalma makes it very easy to write those workflow applications by letting you write it as an ordinary procedural program without any boilerplate.

Swish is an open source web services-based workflow API/Engine. The name Swish is an acronym derived from the phrase Simple Web services Interface to Shark. Swish provides a convenient web services layer on top of Enhydra Shark, offering two key benefits: 1. It opens Shark to a wider range of developers by supporting both Java and non-Java clients. Java clients are supported natively and can use Swish in 'embedded' mode in addition to the web services alternative. 2. It greatly simplifies access to the more commonly-used features of Shark.

BpmScript is a continuation based scripting platform for Business Process Management. Features include:

* Clustered processes
* Worklist Support
* ServiceMix JBI integration
* Web Management Console
* Versioned Processes
* Child Processes
* Remote API
* Half Async Processes
* Parallel Processes

4.2.3 DSS

This section is an extended list Java tools directly or indirectly connected to DSSs (general development environments, inference engines, reasoners and frameworks) some of which used by ISTI.

Editors

Altova's SemanticWorks

Altova's SemanticWorks is a visual RDF/OWL editor from the creators of MLSpy.

Arity's LexiLink

LexiLink is tool for building, curating and managing multiple lexicons and ontologies in one enterprise-wide Web-based application. The core of the technology is based on RDF and OWL.

DERI Ontology Management Environment (DOME)

DOME comprises tool support for Editing & Browsing, Versioning & Evolution, as well as Mapping & Merging, offered in the form of freely combinable Eclipse Plug-ins.

Graphl

Graphl is a tool for collaborative editing and visualisation of RDF graphs.

GrOWL

GrOWL provides a graphical browser and an editor of OWL ontologies that can be used stand-alone or embedded in a web browser.

IBM's IODT

IODT, IBM's toolkit for ontology-driven development.

Intellidimension's RDF InferEd

Intellidimension's RDF InferEd is a powerful authoring environment that gives you the ability to navigate and edit RDF (Resource Description Framework) documents

IsaViz

IsaViz is a visual authoring tool for browsing and authoring RDF models represented as graphs. Developed by Emmanuel Pietriga of INRIA and formerly of W3C and Xerox Research Centre Europe.

Language & Computing's LinKFactory

Language & Computing's LinKFactory is an ontology management tool, it provides an effective and user-friendly way to create, maintain and extend extensive multilingual terminology systems and ontologies (English, Spanish, French, etc.). It is designed to build, manage and maintain large, complex, language independent ontologies.

Metatomix M3t4.Studio Semantic Toolkit
The M3t4.Studio Semantic Toolkit is a free set of Eclipse plugins to allow developers to create and manage OWL ontologies and RDF documents.

Model Futures OWL Editor
The Model Futures OWL Editor is a free tool that is easy to use and install. It offers a simple tree structure user interface and can handle very large OWL files. It also has XMI, Thesaurus Descriptor, and ErWin(TM) import capabilities, and can export ontologies as MS Word(TM) documents.

pOWL
pOWL delivers a PHP and web-based ontology editing and management solution.

RDFe
RDFe is a Schema-Aware RDF Editor, based on pyrple.

Stanford's Protégé
Stanford University's general Protégé 2000 ontology editor tool has a plugin architecture that allows the development of a number of Semantic Web related tools. Examples are an OWL plugin (called Protégé-OWL) to edit RDF and OWL ontologies as well as SWRL rules, a visual editor for OWL (called OWLViz), storage backends to Jena and Sesame, as well as an OWL-S plugin, which provides some specialized capabilities for editing OWL-S descriptions of Web services.

SweDE
SweDE is an open source development environment integrated in Eclipse.

SWOOP
SWOOP from the University of Maryland is a Hypermedia-based Featherweight OWL Ontology Editor, now is under the Google code initiative.

Storing platforms

Cerebra Server
Cerebra Server is a technology platform that is used by enterprises to build model-driven applications and highly adaptive information integration infrastructure.

Cypher
Cypher generates the .rdf (RDF graph) and .serql (SeRQL query) representation of a plain language input, allowing users to speak plain language to update and query databases.

IBM's Web Ontology Manager
IBM's Web Ontology Manager is a lightweight, Web-based tool for managing ontologies expressed in Web Ontology Language (OWL).

IBM Semantic Layered Research Platform
IBM SLRP is a family of open-source Semantic Web software components including an enterprise RDF store, query engine, web application framework, RCP development libraries, and more.

OpenLink's Data Spaces Platform
OpenLink Data Spaces (ODS) is a distributed collaborative application platform for creating Semantic Web presence in conjunction with Web 2.0 application profiles such as: Weblogs, Wikis, Feed Aggregators, Bookmark Mangers, Discussion Forums, Photo Galleries, Social Networks, and more. It provides transparent access to application data via in-built SPARQL support and incorporation of shared ontologies such as SIOC, FOAF, and Atom OWL. ODS is an application of OpenLink Virtuoso, and is available in Open Source and Commercial Editions.

Profium's Semantic Information Router
Profium's Semantic Information Router (SIR) is a content management system using standardized metadata which improves information reusability and allows the user to process and distribute further information acquired from numerous sources in different formats.

Semantic Web Client
The Semantic Web Client Library represents the complete Semantic Web as a single RDF graph. The library enables applications to query this global graph using SPARQL. To answer queries, the library dynamically retrieves information from the Semantic Web by dereferencing HTTP URIs and by following rdfs:seeAlso links. The library is written in Java and is based on the Jena framework.

Siderean's Seamark Navigator
Siderean's Seamark Navigator provides a powerful view of all the information in your enterprise to your users or customers. Web search pages can be combined with product catalog databases, document servers, and other digital information from both inside and outside the enterprise. It also has SPARQL API to get to the data directly.

Software AG's Enterprise Information Integrator (EII)
EII version 2.1 is a globally available information integration product that uses Semantic Web technology. By dynamically combining the meaning and context of business data with the rules that govern its use, Enterprise Information Integrator provides business leaders with the resources to make faster decisions based on real time information availability.

Teranode's Experiment Design Automation
Teranode's Experiment Design Automation (XDA) software is a powerful platform that allows scientists to automate lab and in-silico experiments and manage data within and across laboratories, to improve the speed and quality of R&D projects.

Thetus Publisher
Thetus provides knowledge discovery and modelling infrastructure software that enables organizations to describe, structure, search, relate, model, share, and re-use information—independent of schema or device.

Top Quandrant's TopBraid Composer
Top Quandrant's TopBraid Composer is a complete standards-based platform for developing, testing and maintaining Semantic Web applications. The tool also implements RDFa and GRDDL.

VisualKii
VisualKii is a multi-purpose visual programming platform based on Java. It has libraries for processing RDF, N3 and N-TRIPLES models by visually defining data flow and arranging processing steps. It also includes support for SPARQL queries.

@Semantics' Enterprise Information Integration
@Semantics Enterprise Information Integration (EII) is a federated approach to data management. The EII approach rely fully on open standards, using RDF/S for describing information.

RDF Triple Store Systems
The Aduna Metadata Server automatically extracts metadata from information sources, like a file server, an intranet or public web sites. The Aduna Metadata Server is a powerful and scalable store for metadata. The Metadata Server is based on the Sesame server.

Boca
Boca enterprise RDF store is a Java-based store and client libraries which features named-graph-based RDF storage, access controls, versioning, replication and local persistence for offline access, and notifications (eventing) to distributed clients. Boca is part of IBM Semantic Layered Research Platform (SLRP).
D2RQ and D2R Server
D2RQ is a Java library that provides access to the content of relational databases through SPARQL, the Jena API, and the Sesame API. D2R Server is a SPARQL and RDF server based on D2RQ.

Franz Inc's AllegroGraph
AllegroGraph is a system to load, store and query RDF data. It includes a SPARQL interface and RDFS reasoning. It has a Java and a Prolog interface.

Intellidimension's RDF Gateway
Intellidimension's RDF Gateway is an RDF Triple database with RDFS reasoning and SPARQL interface.

Jena's Joseki The Jena RDF Server
Jena's Joseki layer offers an RDF Triple Store facility with SPARQL interface Joseki is a server for publishing RDF models on the Web. Models have URLs and they can be access by HTTP GET. Joseki is part of the Jena RDF framework.
Joseki is an HTTP and SOAP engine supports the SPARQL Protocol and the SPARQL RDF Query language. SPARQL is developed by the W3C RDF Data Access Working Group.

Joseki Features:
RDF Data from files and databases
HTTP (GET and POST) implementation of the SPARQL protocol
SOAP implementation of the SPARQL protocol

Kowari
The Kowari Metastore is an Open Source, massively scalable, transaction-safe, purpose-built database for the storage and retrieval of RDF, written in Java. Kowari has not been maintained since December, 2005.

Mulgara
The Mulgara Semantic Store is an Open Source, massively scalable, transaction-safe, purpose-built database for the storage and retrieval of RDF, written in Java. It is an active fork of Kowari.

OpenLink Virtuoso
Virtuoso a SQL-ORDBMS and Web Application Server hybrid (aka Universal Sever) that provides SQL, XML, and RDF data management in a single multithreaded server process. Triple Store access is available via: SPARQL, SIMILE Semantic Bank API, ODBC, JDBC, ADO.NET, XMLA, WebDAV, and Virtuoso/PL (SQL Stored Procedure Language). The product is available in Open Source and Commercial editions.

Oracle Spatial 10g
Oracle Spatial 10g includes an open, scalable, secure and reliable RDF management platform. Based on a graph data model, RDF triples are persisted, indexed and queried, similar to other object-relational data types. The Oracle 10g RDF database ensures that application developers benefit from the scalability of Oracle 10g to deploy scalable and secure semantic applications.

OWLIM
OWLIM is a high-performance semantic repository, packaged as a Storage and Inference Layer (SAIL) for the Sesame RDF database.

RDFStore
RDFStore is an RDF storage with Perl and C API-s and SPARQL facilities RAP's RDF server. The RDF server of the PHP RAP environment.

SemWeb for .NET
SemWeb supports persistent storage in MySQL, Postgres, and Sqlite; has been tested with 10-50 million triples; supports SPARQL.

Sesame
Sesame is an open source RDF database with support for RDF Schema inferencing and querying. It offers a large scale of tools to developers to leverage the power of RDF and RDF Schema.

Tucana Suite
Northrop Grumman's Tucana Suite is an industrial quality version of the Kowari Metastore.

YARS
YARS (Yet Another RDF Store) is a data store for RDF in Java and allows for querying RDF based on a declarative query language, which offers a somewhat higher abstraction layer than the APIs.

3Store
3Store is a MySQL based triple store. The server software itself does not expose any interfaces directly to the user, but it can be queried by a number of services, including a column based view and a direct RDF browser.

Multilanguage environments

Euler
Euler is an inference engine supporting logic based proofs. It is a backward-chaining reasoner enhanced with Euler path detection. It has implementations in Java, C#, Python, Javascript and Prolog. Via N3 it is interoperable with W3C Cwm.

Redland RDF Application Framework
The Redland RDF Application Framework is a set of free software libraries that provide support for RDF. It provides parser for RDF/XML, Turtle, N-triples, Atom, RSS; has a SPARQL and GRDDL implementation, and has language interfaces to C#, Python, Obj-C, Perl, PHP, Ruby, Java and Tcl.

C++ Developers

Brahms

Brahms is a fast main-memory RDF/S storage, capable of storing, accessing and querying large ontologies. It is implemented as a set of C++ classes.

Java Developers

Corese

Corese Corese stands for Conceptual Resource Search Engine. It is an RDF engine based on Conceptual Graphs (CG).
It enables the processing of RDF Schema and RDF statements within the CG formalism, provides a rule engine and a query engine accepting the SPARQL syntax.

DartGrid

DartGrid is an java-based application development framework for integrating heterogeneous relational databases using semantic web technologies.

Jena

Jena Java RDF API and toolkit is a Java framework to construct Semantic Web Applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine.

Jena is open source and grown out of work with the HP Labs Semantic Web Programme.

The Jena Framework includes:

- A RDF API
- Reading and writing RDF in RDF/XML, N3 and N-Triples
- An OWL API
- In-memory and persistent storage
- SPARQL query engine

Jena is probably the most widely used Java APIs for RDF and OWL, providing services for model representation, parsing, database persistence, querying and some visualization tools. Protege-OWL editor always had a close relationship with Jena. The Jena ARP parser is still used in the Protege-OWL parser, and various other services such as species validation and datatype handling have been reused from Jena. Protege-OWL is closer integrated with Jena: this integration allows programmers to user certain Jena functions at run-time, without having to go through the slow rebuild process each time.

Jena provides broad types of rules and usages, in particular:

1. deductive inference rules (i.e.: if A and B are true then C is also true) used for RDFS, OWL inference

2. transformation rules (i.e.: if see structure A then output structure A') used for mapping between different schemas/ontologies
3. constraint rules (i.e.: A and B can't both be true) used for validating consistency of OWL instance data, integrity checking
4. condition/action rules (i.e.: if you see pattern A then do action B) implemented in Jena by wrapping action B as a "builtin"

Three modality of running rules are available in Jena: forward, backward and hybrid

In the forward mode, efficient when all the deductions are needed, first time inference model is queried all rules will run until no more can fire, results are stored in a deductions model, future queries fast new adds to base model processed .
In the backward mode, that is useful only if querying for a small part of the set of possible deductions: when query is made to inference model only those rules needed to compute the query results fire; future queries will (may) trigger inference again.
In the hybrid mode, both rule engines can be used together: forward rules can create new backward rules; forward rules can't see results of backward rules.

The built-in Jena rule-based reasoners are able to provide semantic entailments for ontologies using OWL-lite, and some constructs from OWL-DL and OWL-full. However, for large or complex ontologies the rule-based approach can computationally very expensive. Moreover, there are useful modelling primitives in OWL-DL that many semantic-web applications will wish to make use of, but which the rule-based reasoners don't handle. Jena's architecture provides a mechanism for attaching external reasoners to Jena models. This is realized through a transparent gateway between Jena ontology models and external reasoners implementing the DIG description logic reasoner interface.
The DIG interface is an emerging standard for providing access to description-logic reasoning via an HTTP-based interface to a separate reasoning process. Available DIG reasoners are Racer, FaCT, Bossam and Pellet. In principle, Jena should work with any conformant DIG reasoner .
Recommendation of Jena developers is to use the Pellet reasoner, since it is robust and scalable, and is available under an open-source license. We have tested both Pellet and Bossam with good results.
The Jena2 persistent storage subsystem implements an extension of the Model class that provides transparent persistence for models through the use of a database engine. Three database engines are currently supported, MySQL, Oracle, PostgreSQL and Microsoft SQL server, on both Linux and WindowsXP.
Jena has also the ability to be used as an RDF database via its Joseki layer.

Jess

It is a rule engine and scripting environment written in Java. It provides to Java developers to call a rule engine based on declarative rules. Furthermore it provides a scripting language in order to gives access to all of Java's APIs.

Jess uses an enhanced version of the Rete algorithm to process rules. Rete is a very efficient mechanism for solving the difficult many-to-many matching problem. Its rule engine works with the backward modality.

Unfortunately ***Jess*** is a commercial product. A free version is available only for academic use.

JRDF

JRDF Java RDF Binding is an attempt to create a standard set of APIs and base implementations to RDF using Java.

Mandarax

It is an open source java class library for deduction rules. It provides an infrastructure for defining, managing and querying rule bases.

Mandarax is based on backward reasoning. This fits perfectly in a computing landscape based on a pull model (e.g. a transaction initiated from a web site). Data (e.g., from relational databases) can be integrated on the fly at query time, no replication is necessary.

The easy integration of all kinds of data sources. E.g., database records can be easily integrated as sets of facts and reflection is used in order to integrate functionality available in the object model. Other data sources (like EJB, data returned by web services etc) can be integrated as well.

Mandarax contains a reference implementation of an inference engine. This engine is very flexible: unification algorithm, loop checking algorithm and selection policy can be configured.

Rule bases can be made persistent using the XKB module. This module stored rules and other knowledge in a format similar to RuleML. The Mandarax team itself is part of the RuleML initiative working on a XML standard for rules. Export and import of RuleML rule bases is supported.

Mandarax is free and open source. The software license used is the GNU lesser general public license, making the software suitable for both open-source and commercial projects.

Knowledge bases can be queried as relational databases using a specific jdbc driver.

Extensions are available including graphical user interface components (swing and servlet/jsp tag based) and other add-ons.

Unfortunately Mandarax appears no more developed since two years.

OWLJessKB

OWLJessKB is a description logic reasoner for OWL. The semantics of the language is implemented using Jess, the Java Expert System Shell. Currently most of the common features of OWL lite, plus some and minus some.

RDFSuite
ICS-FORTH RDFSuite open source, high-level scalable tools for the Semantic Web. This suite includes Validating RDF Parser (VRP), a RDF Schema Specific DataBase (RSSDB) and supporting RDF Query Language (RQL).

YARS
YARS (Yet Another RDF Store) is a data store for RDF in Java and allows for querying RDF based on a declarative query language, which offers a somewhat higher abstraction layer than the APIs of RDF toolkits such as Jena or Redland.

SWEETRULES
SWEETRULES it is an open source rule framework. It translates between various rule formats and it executes SWRL and RuleML using a variety of rule engines such as IBM Common Rules, XSB Prolog, JESS, Jena 2.

Python Developers

CWM
Closed World Machine (CWM) data manipulator, rules processor and query system mostly using the Notation 3 textual RDF syntax. It also has an incomplete OWL Full and a SPARQL access

Pyrple
pyrple parses RDF/XML, N3, and N-Triples. It has in-memory storage with API-level querying, experimental marshalling, many utilities, and is small and minimally interdependent. It can do graph isomorphism testing, rule application, etc.

RDFLib
RDFLib, an RDF libary for Python, including a SPARQL API. The library also contains both in-memory and persistent Graph backends.

4Suite 4RDF
The 4Suite 4RDF an open-source platform for XML and RDF processing implemented in Python with C extensions.

OWL Reasoners

Bossam
Bossam is a rule-based OWL reasoner (free, well-documented, closed-source) for the semantic web. It is basically a RETE-based rule engine with native supports for reasoning over OWL ontologies, SWRL ontologies, and RuleML rules. Additionally, Bossam includes several expressivity features including: 1) URI references as symbols, 2) 2nd-order logic syntax, 3) disjunctions in the antecedent

and conjunctions in the consequent (both via Lloyd-Topor transformation), 4) URI-based java method attachment, 5) support for both negation-as-failure and classical negation.
You can use Bossam for loading, inferencing, and querying over the set of documents. The set can include any combination of the following documents.

a. RDF(S) documents (in RDF/XML or in N3)
b. OWL documents (in RDF/XML or in N3)
c. Bossam rule documents
d. SWRL(+OWL) documents (in OWLX or in RDF/XML)

Also, you can call Java objects from the antecedent or consequent of rules through the URI-based java method attachment, thus enabling you to mix Java objects into the combination of rules and ontologies.
With Bossam, it's possible to introduce Java method calls into SWRL rules.

FaCT++
FaCT++ is an OWL DL Reasoner implemented in C++.

KAON2
KAON2 is an infrastructure for managing OWL-DL, SWRL, and F-Logic ontologies. it is capable of manipulating OWL-DL ontologies; queries can be formulated using SPARQL.

Pellet
Pellet is an open-source OWL-DL reasoner written in Java, originally developed at the University of Maryland's Mindswap Lab, and funded by a diverse group of organizations.
It can be used in conjunction with both Jena and OWL API libraries; it can also be downloaded and be included in other applications. Pellet is based on the tableaux algorithms developed for expressive Description Logics (DL). It supports the full expressivity OWL-DL including reasoning about nominals (enumerated classes). In addition to OWL-DL, as of version 1.4, Pellet supports all the features proposed in OWL 1.1, with the exception of n-ary datatypes.
Thus the expressivity of supported DL is SROIQ(D), which extends the well-known DL SHOIN(D) (the DL that corresponds to OWL-DL) with

- qualified cardinality restrictions,
- complex subproperty axioms (between a property and a property chain),
- local reflexivity restrictions,
- reflexive, irreflexive, symmetric, and anti-symmetric properties,
- disjoint properties, and
- user-defined datatypes.

Pellet provides many different reasoning services as described below. It also incorporates various optimization techniques described in the DL literature and contains several novel optimizations for nominals, conjunctive query answering, and incremental reasoning.
Pellet supports SWRL DL-Safe rules.

RacerPro
RacerPro is an OWL reasoner and inference server for the Semantic Web

Prolog Developers

Dlpconvert
dlpconvert is a tool for the conversion of the Horn fragment of OWL (called DLP) from XML or RDF syntax to Prolog syntax.

SWI-Prolog
SWI-Prolog is a comprehensive Prolog environment, which also includes an RDF Triple store. There is also a separate Prolog library to handle OWL.

INTERPROLOG
InterProlog is an open source Java front-end and functional enhancement for standard Prologs, running on Windows, Linux and Mac OS X, and currently supporting the top open source logic engines: XSB Prolog from the USA (most declarative), SWI Prolog from the Netherlands (best environment) and YAP Prolog from Portugal (fastest). InterProlog comes with Prolog term visualization aids and programming examples, namely a graphical Sudoku puzzle editor and solver.

On-line Validators

BBN OWL Validator BBN OWL Validator
OWL Consistency checker OWL Consistency checker (based on Pellet)
WonderWeb OWL-DL Validator WonderWeb OWL-DL Validator
W3C's RDF Validator W3C's RDF Validator
RDF/XML and N3 Validator rdfabout.com's Validator
VIStology's ConsVISor OWL Consistency checker ConsVISor

4.2.4 CDSS

Clinical decision-support systems (CDSS) are generally defined as any computer program designed to help health professionals make clinical decisions. Historically, CDSS primarily were focused on diagnostic recommendations, pediatric decision-support can be provided by any computer system that deals with clinical data and medical knowledge to help deliver patient-specific advice. Clinical decision-support systems can be categorized by type (simple rule-based alerts vs more complex methods like neural networking and Bayesian statistics), domain (problem-focused vs general diagnostic support), or access (handheld computer vs Web-based vs integrated within an electronic medical record [EMR]).
The first computerised clinical decision support systems, used to support health care providers in daily clinical practice, were developed in the 1970s, but even though there have been a number of individual successes since then, their impact on routine clinical practice has not been as strong as expected.
Examples of these first systems are **AAHELP: de Dombal's system** for acute abdominal pain (1972), **INTERNIST-I** for the support of complex diagnosis of complex problems in general internal medicine (1974), **MYCIN** a rule-based expert system designed to diagnose and recommend treatment for certain blood infections (1976). Since 80's some successful systems were commercialized, such as **DX-plain**, a decision support system which uses a set of clinical findings (signs, symptoms, laboratory data) to produce a ranked list of diagnoses which might explain (or be associated with) the clinical manifestations. DXplain provides justification for why each of these diseases might be considered, suggests what further clinical information would be useful to collect for each disease, and lists what clinical manifestations, if any, would be unusual or atypical for each of the specific diseases. **QMR** (Quick Medical Reference) was another commercialized CDSS. QMR was a diagnostic decision-support system with a knowledge base of diseases, diagnoses, findings, disease associations and lab information. QMR was designed for 3 types of use: as an electronic textbook; as an intermediate level spreadsheet for the combination and exploration of simple diagnostic concepts; as an expert consultant program. QMR was developed in 1980 by the University of Pittsburgh and First Databank, California.
More recent clinical decision support systems were developed for very different clinical settings, such as HIV-management or cancer diagnosis and treatment. Some of these systems are actually in routine use and other was commercialized.

LISA (2004), is an information and decision support system, actually in use and under evaluation, for collaborative care in childhood acute lymphoblastic leukaemia.
LISA's main components are:

- A centralised Oracle database holding all patient information (drug schedules, blood and toxicity results, doses prescribed etc.) and accessible by health professionals from different sectors in different locations.
- A web-based decision support module, implemented using PRO*forma* guideline development technology, designed to provide advice about dose adjustments in the treatment of acute childhood lymphoblastic leukaemia.

RETROGRAM (1999) is a HIV decision support system for genotype interpretation, is used in clinical practice and is under continuing development and update. The decision support technology used in the implementation of RetroGram is AREZZO, based on PRO*forma* and developed by *Infer*Med Ltd., London. The medical content was developed by Virology Networks, HIV Pharmacology. RetroGram is jointly owned by Virology Networks and Hoffmann-La Roche.

TherapyEdge HIV is a Web-based decision support system for the treatment of HIV. It forms the first release module of TherapyEdge, a knowledge-based system for the treatment of chronic diseases.
TherapyEdge-HIV is a HIV specific patient record, monitoring and reporting system with built in decision support. TherapyEdge is used in over 36 clinical sites in 32 countries. It is used by NGO's as their primary longitudinal patient record, by government organizations to track program indicators and expenditures, and by clinicians to improve the quality of care. TherapyEdge is used by researchers to learn more about the resistance profiles and by virologists and clinicians who want to improve the quality of care with the decision support tools.
TherapyEdge HIV systems provide different Decision Support Functionalities:

- Integration of viral resistance interpretation
- Drug-Drug Interactions
- Drug-Condition Interactions
- Alerts and Messaging management

TherapyEdge HIV graphically tracks and automatically processes a patient's clinical data – including medical conditions, medications, genetic tests for drug resistance, drug efficacy and toxicity data – through an extensive knowledge base of pharmacological and clinical information created and maintained in collaboration with expert HIV clinicians and researchers. The system utilizes an AI engine and knowledge base to assess a patient's current status and generate patient-specific, optimized treatment alternatives for a clinician to review and compare. In this way, the system can be used to generate comprehensive, individualized treatment plans for patients. TherapyEdge is available to subscribers via the Internet.
TherapyEdge's real-time, intelligent alerting system automatically checks for drug interactions, medical conditions or side-effect issues, as well as abnormal lab results and drug dosing. Additionally, the system de-identifies patient data and provides a longitudinal database that can be queried for drug efficacy data, clinical

outcomes and quality of care information. The system therefore supports the validation of effectiveness of health services and therapeutic interventions.
Version (3.2) of TherapyEdge-HIV was released by ABL, SA in early 2005. The new system incorporates a data warehouse for aggregation of data from on-site installations of TherapyEdge-HIV. There is a PDA data capture module, Rapid Capture Forms, for sites seeing large numbers of HIV patients; a patient interface, and advanced reporting tools (that include USAID and WHO Reports.)
The AI Therapy Evaluation Report has been updated to support treatment guidelines from numerous contries as well as localized drug formularies. ABL offers an HL7/XML module to communicate with other systems.

ATHENA (Assestment and Treatment of Hypertension: Evidence-Based Automation) is a recently developed CDSS for the cardiovascular clinical domain, supporting for the management of hypertension in primary care (2002). The ATHENA system implements guidelines for hypertension using Stanford Medical Informatics EON architecture. ATHENA encourages blood pressure control and recommends guideline-concordant choice of drug therapy in relation to comorbid diseases. ATHENA has an easily modifiable knowledge base that specifies eligibility criteria, risk stratification, blood pressure targets, relevant comorbid diseases, guideline-recommended drug classes for patients with comorbid disease, preferred drugs within each drug class, and clinical messages. Because evidence for best management of hypertension evolves continually, ATHENA DSS is designed to allow clinical experts to customize the knowledge base to incorporate new evidence or to reflect local interpretations of guideline ambiguities. Together with its database mediator, Athenaeum, ATHENA DSS has physical and logical data independence from the legacy Computerized Patient Record System (CPRS) supplying the patient data, so it can be integrated into a variety of electronic medical record systems.
In 2006, a second ATHENA DSS prototype application - for Opioid Therapy - was developed. The application implemented parts of the VA/DoD Clinical Practice Guideline (CPG) for the Management of Opioid Therapy for Chronic Pain (VA/DoD, 2003).

4.2.5 Guideline implementation tools

Actual guideline implementation methods employ different representation formalisms and computational techniques, for example:

- Rule-based
- Logic-based
- Network-based
- Workflow (Petri Nets)

Other current methods for representing clinical knowledge take a different approach by structuring guidelines on the basis of XML guideline document models. Methods in this category include GEM, Stepper and HGML.
The DeGeL (Digital electronic Guideline Library) infrastructure of methods and tools represents a hybrid approach: it is capable of supporting both markup and formal, executable methods of representating clinical guidelines.

In this section different approches, methods and tools for implementing guidelines will be considered.

The Arden Syntax

The first version of the Arden Syntax was developed in 1989 as an open standard for the procedural representation and sharing of medical knowledge among different institutions. The **Arden Syntax for Medical Logic Systems** encodes medical knowledge in knowledge base form as Medical Logic Modules (MLMs). An MLM is a hybrid between a production rule (i.e. an "if-then" rule) and a procedural formalism. Each MLM is invoked as if it were a single-step "if-then" rule, but then it executes serially as a sequence of instructions, including queries, calculations, logic statements and write statements. The Arden Syntax makes knowledge portable, but MLMs developed for one environment are are not easily embeddable within another.
The Arden Syntax is not designed for complex guidelines that for example address treatment protocols, it focuses on the sharing of simple modular and indipendent guidelines as clinical alerts and reminders, interpretations, diagnoses, screening for clinical research studies, quality assurance functions, and administrative support reminders. With an appropriate computer program (known as an event monitor), MLMs run automatically, generating advice where and when it is needed, e.g. to warn when a patient develops new or worsening kidney failure.
Regarding the representation primitives, the Arden Syntax has a *logic slot* which is used to encode the decision criteria of a Medical Logic Module (*decision* primitive), and an *action slot* which is used to encode the clinical task that should be performed (*action* primitive). Sequencing tasks can be modelled by chaining a sequence of MLMs. The Arden Syntax provides a definition for the interface to patient data with a *data slot*. However, it does not support patient data modelling. Patient data encoding, which is enclosed within a pair of curly braces ({}), is left to the local site to implement and integrate. This problem, known as *"curly braces problem"*, is one of the major hindrances for the sharability of the MLMs.
Another potential limitation of Arden Syntax is that it does not explicitly define notification mechanisms for alerts and reminders. Instead, this is left to local implementation and is, like database queries, contained in curly braces in a MLM. Explicit notification mechanisms in the Syntax itself may be a part of a future edition.

The Arden Syntax has been used by different institutions and companies to develop and implement guidelines in multiple clinical settings. Most commercial

applications incorporating MLMs are developed by individual vendors primarily for use within their own environments. Vendors who have developed Arden-compliant decision support applications include:

- Eclipsys Corporation
- McKesson Information Solutions
- Siemens Medical Solutions Health Services Corporation
- MICROMEDEX (MICROMEDEX® Medical Logic Modules).

Healthcare organisations with Arden-compliant commercial systems include:

- Alamance Regional Medical Center, Burlington, NC (Eclipsys)
- Sarasota Memorial Hospital, Sarasota FL (Eclipsys)
- Columbia-Presbyterian Medical Center, New York, NY (developed with IBM)
- JFK Medical Center, Edison, NJ (McKesson)
- Covenant Health, Knoxville, TN (McKesson)
- St. Mary's Hospital, Waterbury, CT (McKesson)
- Mississippi Baptist Health Systems, Jackson, MS (McKesson)
- St. Vincent's Hospital, Birmingham, AL (McKesson)
- St. Mary's Medical Center, Knoxville, TN (McKesson)
- Meridian Health Systems / Jersey Shore Medical Center, Neptune NJ (Siemens)
- Ohio State University, Columbus OH (Siemens)

Asbru

Asbru is a guideline representation formalism, developed at Standford University and the Vienna University of Technology and is part of the **Asgaard project**. The aim of the Asgaard/Asbru project is to design a set of tasks that support the design and the execution of skeletal plans by a human executing agent other than the original plan designer. During the design phase of plans, Asbru allows to express durative actions and plans caused by durative states of an observed agent. During the execution phase, Asbru allow flexibility to instantiate the skeletal plans with time-oriented patient data. The intentions underlying these plans are represented explicitly as temporal patterns to be maintained, achieved or avoided. The inputs are to the planner are time-oriented raw data (e.g., patient data) and raw protocols (e.g., clinical protocols). The data are acquired from electronic devices or entered by the users. The protocols will be represented and transformed in a sharable representation (the sharable skeletal plan library). The reasoning, based on the sharable plan-library and on the interaction with the user's needs and tasks, results in context-sensitive support, like recommendations, explanations, or visualization of the available and necessary data.

The Asbru language is a task-specific and intention-based plan representation language to embody clinical guidelines and protocols as *time-oriented skeletal plans*. Skeletal plans provide a powerful way to reuse existing domain-specific procedural knowledge, while leaving room for execution-time flexibility to achieve particular goals. In order to manage these, often complex, skeletal plans, Asbru uses a representation of high-level goals (intentions), a representation of

temporal patterns and time annotations, and the development of user interfaces to visualize developed plans. Asbru enables the intentions and goals of a guideline and the temporal dimensions and uncertainties to be defined as an intrinsic part of that guideline. This supports the appropriate application of a guideline in practice and the quality assessment of its application.
Asbru has *conditions* and *preferences* corresponding to decisions as well as *plans* and *plan states* to actions and execution states. Asbru represents temporal constraints, such as *sequence* or *concurrence*, in two dimensions, i.e. ordering constraints, which can take values *parallel*, *any order* or *total order*, and continuation condition, which can take on the values *all completed* or *some completed*. Asbru also provides a third category of temporal constraints, represented as a *cycle*. All of these temporal constraints are represented within is *plan-body*.

Main features of Asbru:

- Prescribed actions and states can be continuous;
- Intentions, conditions, and world states are temporal pattern;
- Uncertainty in both temporal scopes and parameters can be flexibly expressed by bounding intervals;
- Plans might be executed in sequence, all plans or some plans in parallel, all plans or some plans in a particular order, or periodically;
- Particular conditions are defined to monitor the plans' execution;
- Explicit intentions and preferences can be stated for each plan separately. (Intentions are temporal-pattern constraints, e.g., a process intention to administer regular insulin twice a day; an outcome intention to maintain fasting blood glucose within a certain range over at least 5 days a week, that are allocated individual weights signifying their relative importance. Such knowledge is necessary to determine, for instance, whether a care provider is still following most of the guideline, or, at least, its spirit. Such a provider might be applying the guideline in modified fashion, as is, in fact, the case in 50% of inspected guideline applications.)

EON

The EON, developed at Standford University, is a component-based architecture used to build decision support systems that reason about guideline-directed care. The EON architecture consists of several components that facilitate the acquisition and execution of clinical guidelines.
The central research questions of the EON system include:

- How to model clinical guidelines and protocols to provide patient-specific decision support
- How to represent and reason with time-oriented patient data
- How to present and explain decision-support recommendations and conclusions
- How to create a knowledge-engineering environment for easy encoding of guidelines and protocols

EON includes an extensible suite of models to represent parts of a clinical practice guideline, domain ontologies, a view of patient data (virtual medical record), and other entities (e.g. those that define roles in an organization). The guideline model, called **Dharma**, is object-oriented and consists of classes that describe guideline entities as a sequence of structured temporal steps. Dharma is not a monolithic model, meaning that the guideline representation can be extended with additional classes that capture new guideline behaviour. The formalism defines guideline knowledge structures such as eligibility criteria, abstraction definitions, guideline algorithm, decision models, and recommended actions. Besides the Dharma guideline model, the EON architecture also contains run-time components, used to construct guideline execution systems. These systems obtain patient data through a specified temporal database manager or from user input, and then generate recommendations according to the contents of the specific guideline.
EON defines an *activity*, which is a *continuos process*, from an *action*, which is an *instantaneous process*. It offers both a *patient scenario*, used to describe the patient state with regard to decisions made and actions completed, and an *activity state*, used to describe the patient state with regard to the status of activities.
Regarding the temporal constraints representation, EON use a flowchart, and supports nesting through inclusion of subguideline.

Encoding of EON guidelines is done in the **Protégé-2000** knowledge-engineering environment. The encoding process is facilitated by specialized views of the EON guideline model designed to satisfy specific requirements of different classes of guidelines. Theses requirements are conceptualised in terms of a set of guideline tasks, e.g. decision making, specification of work to be performed, interpretation of data, setting goals. A guideline developer using EON creates specialized views of the guideline model by selecting modelling solutions to these tasks.
In the EON guideline model, conditional goals (e.g. if patient is diabetic, the target blood pressures are 135/80) are associated with guidelines and subguidelines. The guideline algorithm is represented as a set of scenarios, action steps, decisions, branches, synchronisation nodes connected by a "followed-by" relation. EON provides three criteria languages to allow usability and medical expressivity:

- A simple object-oriented language that clinicians can use to encode the majority of decision criteria
- A temporal query and abstraction language
- First-order predicate logic.

Among its benefits, EON supported the reusability of medical domain knowledge, temporal queries and abstractions.

Gaston

Gaston is a generic architecture for the design, development, validation and implementation of guideline-based medical decision support systems. The framework facilitates the development and implementation of computer-interpretable guidelines and guideline-based decision support systems. Gaston has

been developed by a joint effort of the Department of medical Informatics of Maastricht University and the Signal Processing Systems group of the Eindhoven University of Technology.
The overall goal of this approach is to improve the acceptance of computer-interpretable guidelines and decision support systems in daily care by facilitating all phases in the guideline development process.

The central research questions of the Gaston project are:

- How to represent and share various types of guidelines using a formal and unambiguous representation.
- How to translate guidelines from a textual format into this formal representation.
- How to handle local adaptation and synchronization between (inter)national and local guidelines.
- How to evaluate guidelines and decision support systems in daily practice.
- How to interface guideline-based decision support systems with external patient information systems.
- How to provide decision support to a care provider in daily practice.
- How to handle real-time patient data sources such as patient monitors that contain streaming physiological data (e.g., heart rate or ECG).

The Gaston framework consists of

5. A guideline representation formalism that uses the concepts of primitives, Problem-Solving Methods (PSMs) and ontologies to represent guidelines of various complexity and granularity and different application domains;
6. A guideline authoring environment that enables guideline authors to define guidelines;
7. A guideline execution environment that translates defined guidelines into a more efficient representation, which can be read in and processed by an execution-time engine.

The guideline representation formalism uses a frame-based model as an underlying mechanism. The formalism is non-monolithic, meaning that it can be extended with additional classes to capture new guideline characteristics.
The Gaston guideline authoring environment represents and visualizes guidelines by temporally sequenced graphs (flowcharts) of frame instances from the guideline model. Guidelines in Gaston consists of various layers (depending on the guideline's complexity or application domain) that describe the control structure of a guideline (flow), its contents (e.g., actual decisions or actions), possible local adaptations, and communication/implementation details (e.g., the method of acquiring patient data or the form of decision support).
The Gaston approach defines various methods for the detection of various logical and procedural errors in guidelines. In addition, the framework also contains a simulation environment where developed guidelines and decision support systems can be tested and evaluated.
Finally, the framework contains a guideline execution environment that is able to execute guidelines and interface with external patient information systems. The

execution environment consists of a core guideline execution engine, which can be extended with additional components (plugins) to communicate with patient information systems, medical databases and patient monitors.

GEM

GEM (Guideline Element Model) is intended to facilitate the translation of natural language guideline documents into a standard computer interpretable format. It encodes considerable information about guideline recommendations in addition to the recommendations themselves, including the reason for each recommendation, the quality of evidence that supports it, and the recommendation strength assigned by the developers. GEM-encoding of guideline knowledge is pursued through a markup process that does not require programming knowledge.
The GEM Cutter application is an XML editor that facilitates guideline markup.
GEM is intended to be used throughout the entire guideline lifecycle to model information pertaining to guideline development, dissemination, implementation, and maintenance. Information at both high and low levels of abstraction can be accommodated. GEM preserves the intent of guideline developers by marking up the actual guideline language.
In 2002, GEM was successfully balloted as an international ASTM standard (no. E2210-02) for the representation of practice guidelines in XML format.

GLARE

GLARE is a domain-independent system for acquiring, representing and executing clinical guidelines. GLARE has been under development since 1997 by the Dipartimento di Informatica, Università del Piemonte Orientale "Amedeo Avogadro", Alessandria, Italy, in co-operation with the Laboratorio di Informatica Clinica, Azienda Ospedaliera S. Giovanni Battista, Torino, Italy.
The system is based on a modular architecture, which includes an acquisition tool and an execution tool.
The GLARE representation language is designed to achieve a balance between expressiveness and complexity. The formalism consists of a limited, but very focused and clearly understandable set of primitives. It is made up of different types of actions: plans (i.e. composite actions, hierarchically decomposable in their sub-actions) and atomic actions. Atomic actions can be queries, decisions, work actions and conclusions. All actions are linked by control relations (e.g. sequence, alternative, repetition), defining their order of execution.
GLARE provides expert physicians with an "intelligent" guideline acquisition interface. This provides different types of checks to help develop a consistent guideline: syntactic and semantic tests verify the "well-formedness" of a guideline. Further, extended Artificial Intelligence (AI) temporal reasoning techniques are used to check the consistency of temporal constraints imposed between actions.

GLIF

GLIF (GuideLine Interchange Format) was developed by Intermed Collaboratory including researchers at Columbia University, Harvard University and Standford

University and was first published in 1998. GLIF was developed to model guidelines through flowcharts that consist of structured scheduling steps, representing clinical actions and decisions, by modelling guidelines in such a manner that they are understandable by human experts as well as by automatic parser used in different clinical decision support systems. The intended purpose of GLIF is to support sharing of computer-interpretable clinical guidelines across different medical institutions and system platforms. GLIF defines an ontology for representing guidelines, as well as a medical ontology for representing medical data and concepts. Tools are under development to support guideline authoring and execution.

The first version of GLIF (GLIF2) facilitated the description of more complex guidelines than other approach (like Arden Syntax) did, but it needed improvement in a number of areas. For example GLIF2 had no constructs that formally allowed the mapping of patient data elements in the guideline onto elements that are used in clinical systems such as EPR (Electronic Patient Record) systems. Furthermore, the constructs that supported alternative decisions, iterations, patient states exceptions and events was lacking. All these issues have been addressed and have recently resulted in the new GLIF3 version.

GLIF has a *decision step*, an *action step* and a *patient state step*, corresponding to decision, action and patient state primitives. GLIF uses a *branch step* and a *synchronization step* to represent sequence in any order and concurrence, while using *the next-step slot* to represent simple sequence. The branch step in GLIF defines a point in a flowchart that is followed by multiple parallel paths or paths that can be traversed in any order. The synchronization step defines a point in a flowchart at which diverged paths converge back, with a continuation criterion defined either as a *boolean criterion*, which is a logical expression of the diverged paths, or as a *k_of_n_criterion*, which is a special case of a Boolean criterion. Since the continuation criterion is a logical expression, GLIF is very expressive in its temporal constraints representation. GLIF also provides nesting through inclusion of subguideline.

In GLIF3 is used an XML-based syntax that represents objects and instances in a text-like manner. Recently the object-oriented expression language **GELLO** (Guideline Expression Language) was proposed to specify decision and eligibility criteria in GLIF. GELLO is a “platform independent” object-oriented query and logical expression language for clinical decision support. GELLO is designed to meet the following requirements:

- the need for a standard, object-oriented language to formulate expressions for manipulating clinical data and providing decision support in various clinical applications.
- the need to support sharing of clinical knowledge between applications in heterogeneous clinical environments.
- the need to be vendor independent, platform independent, object oriented and compatible with the HL7 Reference Information Model (RIM), easy to read and write, free of side-effects, extensible and shareable.

The GELLO language can be used to:

- Build up queries to extract and manipulate data from medical records.

- Construct decision criteria by building up expressions to reason about particular data features/values. These criteria can be used in decision support knowledge bases such as those designed to provide alerts and reminders, guidelines, or other decision rules.
- Create expressions, formulae, and queries for other applications. The query and expression languages share a common Object-Oriented data model since the expressions must operate on data supplied by queries.

Currently, two tools are used to develop the GLIF model in terms of classes and attributes: **Protègè** and **GEODE**. They are also used as knowledge acquisition tools to facilitate the entering of guidelines. Both tools visualize GLIF guidelines by means of flowcharts.

Protégé is a free, open source ontology editor and knowledge-base framework. The Protégé platform supports two main ways of modeling ontologies via the Protégé-Frames and Protégé-OWL editors. Protégé ontologies can be exported into a variety of formats including RDF(S), OWL, and XML Schema. Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development.

The most recent execution engine development is the Guideline Execution Engine (**GLEE**), developed to interpret guidelines encoded in the GLIF3 format and to integrate with clinical information systems for guideline implementation.

At Columbia University, GLEE is being integrated with the Clinical Event Monitor and the computerized physician order entry (CPOE) system to provide clinical decision support. GLEE is also being tested for quality assurance purposes, to examine the potential deviation of the care of specific patients from the standard.

Haifa University is developing:

- a process for local adaptation of GLIF-encoded guidelines
- a mapping ontology from GLIF-encoded guidelines to EMRs.

GUIDE

GUIDE is a component-based multi-level architecture designed in 1998, and re-engineered in 2002, to integrate a formalized model of the medical knowledge contained in clinical guidelines and protocols with both workflow management systems and Electronic Patient Record technologies.

The GUIDE environment integrates three main independent modules:

1. Guideline Management System (GlMS) for providing clinical decision support
2. Electronic Patient Record (EPR)
3. Workflow Management System (WfMS) or Careflow Management System (CfMS) for providing organisational support.

The interaction between GUIDE modules is message-based and defined by specific contracts. The paradigm employed ('Separation of Concerns' - SoC) means that one domain doesn't need to know anything more than current contract details and the shared terminologies and ontologies in order to communicate with another domain. This approach reduces crosstalk and supports parallelization. Further, the strong separation between medical and organizational issues helps

improve interaction between experts of different skills and backgrounds and the system.
The GUIDE model is based on Petri Nets, a formalism invented for modelling concurrent processes. The strength of the formalism, when applied to healthcare, is its ability to support the modelling of complex concurrent processes (sequential, parallel and iterative logic flows). The formalism has been extended to support improved modelling of time, data and hierarchies. The GUIDE model has a *wait action*, which seems not to be a real action, but patient state may still change during this process because the underlying patient pathophysiology status may change over time. GUIDE does not define patient state explicitly but encode it through the Petri nets representation, which implies the existence of a patient state. GUIDE model also provides another important representation feature: nesting is through decomposition of *task*.
GUIDE is applied in management of stroke patients in hospitals; management of breast cancer in hospital and home-care settings; management of patients with heart failure in primary care.

HGML (Hypertext Guideline Markup Language**)**
HGML is an XML/XHTML specification for the identification of condition and recommendation elements within text guidelines. HGML defines tags for conditional and associated recommendation elements which can be identified in a guideline text using a markup editor. The correlation of conditional variables, subject to their constraints, to clinical data allows delivery of recommendations linked to their original context.

A guideline editor - an enhancement of the Mozilla Web Browser's "Composer" feature - has been developed to support guideline markup. An associated Guideline Evaluation Engine allows information about a patient (either from an Electronic Medical Record system, or directly from a clinician) to be collected, and relevant recommendations to be returned to a web browser. The output from the system is linked to the related text in the original guideline, and to supporting evidence including references, related text, and survival curves from models evaluated by Decision Maker, a companion project developed and maintained by members of the HGML development group.

PRESTIGE(*Patient record supporting telematics and guidelines*)
The Prestige project is a cooperative effort at multiple European sites. The project focuses on both guidelines and protocol-based care, as well as on resource management in the daily practice of healthcare. The conceptual model and architecture grew from and extends **DILEMMA** (protocol knowledge representation), **NUCLEUS** (clinical act management) and **GALEN** (concept representation and term manipulation). These are all projects supported by the European Commission for healthcare telematics.
The Prestige Conceptual Guideline Model describes the kind of information - the concepts and the relationships between them - needed to support the use of guidelines in the planning and provision of personal healthcare. This implies a wide-ranging ability to represent knowledge about medicine, patients and carers,

and the enterprises and personnel that care for them. It has proved possible to produce such a model that is portable, i.e. is applicable in any organisation using any technology.
It provides an essential part of a meta-language for those responsible for writing the generic clinical scripts for particular scenarios, the protocol authors.
The Model has two major subdivisions. The first describes healthcare in general, and the second focuses on clinical protocols. The Model has been built and stored using a proprietary object-oriented **CASE** tool, SELECT Enterprise, which can produce skeleton software if required.
Two guideline authoring tools have been built:

- **GAUDI** (Guideline AUthoring and Dissemination Tool), illustrated below, which incorporates a terminology server and model (GRAIL, developed by GALEN);
- **GLEAM** (GuideLine Editing and Authoring Module), which supports direct editing of an application knowledge base.

PRESTIGE is structured into several actions addressing major healthcare specialties, including cardiothoracic disease, neurological disease, diabetes and general practice. Substantial user involvement has been assured through national and specialist user groups and renowned healthcare centres.
PRESTIGE will make available a common pool of interoperable generic technologies to these user groups and clinical sites using the following technologies:

- architectures for multimedia patient records
- act management
- terminology and multilingual services
- generic models for representing clinical guidelines and protocols.

The outcome of PRESTIGE will be an installed and sustainable healthcare telematics infra-structure that supports the dissemination and application of research based and consensus based guidelines that in turn support best practice standards for routine clinical care.

PRODIGY (Prescribing RatiOnally with Decision-support In General-practice studY)
The Prodigy project is funded by the National Health Service of the United Kingdom. The goal of this project is to develop a guideline-based decision-support system to assist general practitioners with patient care. In the Prodigy model, a guideline recommendation is considered to be appropriate for various combinations of patient states (scenarios) given a particular diagnosis; in turn, a scenario may have different strategies for treatment. Each of these strategies can then be further broken down into more detailed instructions.
The model has been under development since 1995 and the most recent version (Prodigy 3) is commissioned to address chronic disease management, such as hypertension, diabetes or angina, in primary care. To facilitate knowledge engineering by domain experts, Prodigy aims at producing the simplest, most understandable guideline model sufficiently expressive to represent chronic disease management guidelines.

The model enables a guideline to be organised as a network of patient *scenarios*, *management decisions* and *action step* which produce further scenarios. Scenarios are patient states defined by the patient's condition and current treatment.
It uses various techniques for managing complexity of guideline models, such as partitioning a clinical algorithm into:

- *management algorithms* that model decisions alternatives;
- *consultation templates* that define context-sensitive best-practice data-collection and evaluation recommendations;
- *subguidelines* that refine scenarios and decision alternatives. The Prodigy model splits clinical interventions into actions, which are instantaneous, and activities that are started and persist until they are modified or stopped.

The PRODIGY3 decision model uses *rule-in* and *rule-out* conditions associated with each available alternative to determine the preferred course of action.
The management over time of a patient according to a guideline specification can be viewed as the traversal of a number of selected *scenarios* and associated actions and further decision points along a single path. Sequencing of actions is achieved by defined *followed-by* relations.

PRO*forma*

PROforma, is a system that has been developed at the Imperial Cancer Research Fund (UK) over the last few years for the general purpose of building decision support and intelligent agents. It provides a formal mechanism for specifying the patient data, medical knowledge, tasks that represent CPGs (Clinical Practice Guidelines), and the constraints among them.
The technology includes the PRO*forma* language, a formal specification language (as that term is used in software engineering), a knowledge representation language (as understood in AI) and a set of Prolog and Java tools for building applications in the language. The basic tasks or objects supported by PROforma include a *plan*, a *decision*, an *action* and an *inquiry step*. In PRO*forma* the notion of a task is central; a guideline application is modelled as a set of tasks and data items. The PRO*forma* task model divides from the keystone (generic task) into four types: plans, decisions, actions and enquiries. A graphical knowledge editor for the creation of CPGs complements the knowledge representation model. The editor performs consistency and completeness checks prior to creating embeddable objects or steps. An enactment engine for testing and execution is also available as a stand-alone application or for use within a clinical information system.
PRO*forma* is essentially a first-order logic formalism extended to support decision making and plan execution, but it also incorporates a number of well known features of non-classical logics (e.g. modal logic, temporal logic) and two novel logics (LA, logic of argument and L_{OT}, logic of obligation and time) to support decision making and action control.

The technology includes a suite of PRO*forma* authoring and execution software that incorporate **CASE** and verification tools. It has been shown to meet specific requirements of medical applications though the language and tools are generic.
PRO*forma* is a continuing area of active research at Cancer Research UK, particularly for safety-critical applications. PRO*forma* is the platform for a number of clinical applications developed by the lab.

Stepper

The Stepper system is *document–centric*: it formalises the initial text in multiple user–definable steps corresponding to interactive XML transformations. The Stepper project has two main goals:

1. To develop a stepwise method for formalization (in this context, XML transformation) of text documents of clinical guidelines
2. To develop the Stepper tool, an XML editor enhanced with features to support the above method

Some guideline formalisation methods favour a one-step, top-down approach using a pre-defined model and the involvement of a domain expert and knowledge engineer. This process can suffer from a lack of transparency as shown, for example, in the difficulty of tracing back the exact source of every piece of knowledge in a resulting guideline application. Stepper, on the other hand, has been designed as a document-centric tool which takes a guideline text as its starting point, and splits the formalisation process into multiple user-definable steps, each of which corresponds to an interactive XML transformation. The result of each step is an increasingly formalized version of the source document.
Stepper builds the guideline model bottom–up, through filling marked–up content into knowledge containers (of pre–defined categories). An advantage of document–centric approaches is better coverage of information that does not fit well into the compact model (esp. broader context of clinical care), and lower risk of leaving an important piece of information unspotted in the original text. Their main disadvantage is however difficulty of bridging the above–mentioned semantic gap: models built bottom–up typically stay on the half–way to fully operational application, since the textual content, though delimited with tags, still has to be interpreted by a human.
Stepper is written in Java and it contains an embedded XSLT processor which carries out non-interactive transformation. Mark-up and iterative transformation are carried out by rules expressed in a new transformation language based on XML called XKBT (XML Knowledge Block Transformation); besides it provides an interface showing the interconnection between the source text and the model.
The Stepper method and tool have been tested in the formalization of WHO/ISH (International Society of Hypertension) hypertension guidelines and in guidelines for breast cancer care. Evaluation has shown that using step-by-step approach makes it possible to transform medical guidelines into fragments of operational code (e.g. Java) or into an instance of medical ontology (such as Asbru). Rules for transforming guidelines into the Asbru language have been implemented.
Although Stepper has been designed to support a bottom-up, stepwise approach to guideline formalisation, the tool is not yet designed to support the development of

a complete decision support application. A series of XML transformations cannot produce a standalone, executable application. A main advantage of Stepper lies in the clarity of the step-by-step approach it fosters. All user activities (all single transformations) are documented in XML files so another user can easily review the whole transformation process.
The document-centric approach supported by the Stepper tool ensures that all additions of "external knowledge" (knowledge not contained in textual guidelines) are kept track of in the guideline model. The approach can also help highlight knowledge missing from a source guideline.

SAGE (Standards-Based (Shareable) Active Guideline Environment)
The SAGE project takes a model-based approach where it is developed a formal model that specify the entities and relationship that constitute a computer-interpretable guideline. The model acts as a template for encoding instances of guidelines. Through the model, the medical knowledge in a guideline is formalized so that electronic applications can apply it to generate patient-specific recommendations for clinical decisions and actions.
The SAGE Guideline Model is designed to

1. use standardized components that are sufficiently granule and precise, to allow interoperability of guideline execution elements with the standard services provided within vendor clinical information systems.
2. encode guideline knowledge needed to provide situation-specific decision support, and to maintain linked explanatory resource information for the end-user
3. include organizational knowledge to capture workflow information and resources needed to provide decision-support in enterprise setting
4. to be well structured, so that the bulk of guideline knowledge can be encoded, through a guideline workbench, by clinicians with basic understanding of the guideline model.
5. synthesize prior guideline modelling work for encoding guideline knowledge needed to provide situation-specific decision support, and to maintain linked explanatory resource information for the end-user

The SAGE's approach to integrating guideline-based decision support with the workflow of care process; the success of clinical decision-support systems (DSSs) depends heavily on how the system is integrated into the care process is widely recognized. The SAGE project takes the approach that, as a provider of decision-support services to CISs, SAGE will not be in control of a host systems' workflow management. Thus, the SAGE modelling approach does not require detailed workflow to be modelled unlike for example the University of Pavia's care flow methodology. Instead, the SAGE system will respond to opportunities for decision support in the care process. We need to model enough of workflow contexts to recognize appropriate events that should trigger decision-support services. Upon receipt of such a triggering event, the SAGE DSS will deliver, through existing functions of the CIS, guideline-based recommendations appropriate for members of a care team. The implication of this approach for guideline modelling is that guideline knowledge must support operations in an event-driven reactive system,

and it must take into account clinical and organization contexts such as care setting and provider roles. Instead of just creating an electronic version of a clinical practice guideline, guideline modelling in SAGE formalizes guideline knowledge being used in specific scenarios and settings.
With Sage it is possible to:

- Represent arbitrarily complex medical knowledge
- Describe event timing and scheduling within clinical workflows
- Provide decision logic for medication choices, dosing, contraindications
- Specify flowsheets as workflow enhancers (nursing, medical, RT, PT, etc.)
- Describe pro-active monitoring of time sensitive clinical events
- Customize evidence-based care to level of individual patient
- Specify alerts and their escalation
- Provide precise specifications for where in the clinical workflow guidance should be activated
- Be aware of theoretical workflows and real active workflows; and can provide triggers for connecting them.
- Specify opportunities for decision support woken up via workflows: fits perfectly with ForeSight modes of operation.

The SAGE project seeks to create a guideline model that:

- uses standardized components that allow interoperability of guideline execution elements with the standard services provided within vendor clinical information systems.
- includes organizational knowledge to capture workflow information and resources needed to provide decision-support in enterprise setting
- synthesizes prior guideline modelling work for encoding guideline knowledge needed to provide situation-specific decision support and to maintain linked explanatory resource information for the end-user

Innovative features of the SAGE guideline model include:

- Organization of guideline recommendations as recommendation sets consisting of either Activity Graphs that represent guideline-directed processes or Decision Maps that represent recommendations involving decisions at a time point.
- Use of a suite of data models and services as interfaces to clinical information systems.
- Systematic use of standard terminologies
- Deployment-driven guideline modelling methodology.

5. Functional Specifications of Data Processing

In this chapter, we describe the representation features relevant to HF domain and we determine, according to the discussion in the previous chapters, the functional specifications of data processing. For sake of clarity, the exposition is split into two parts; namely we discuss separately the functional specifications needed for the routine clinical practice environment and for the long-term research environment. It is understood that both refer to advanced signal processing, touch hot topic in active research areas and may involve new algorithms; nevertheless their aims are substantially different. For routine clinical practice needs refinement of well-established parameters and steps towards their automatic computation, while the research environment would benefit from *novel representation features* to get more insight into heart failure.

5.1 Routine Clinical Practice Environment

Clinical examinations that are part of the routine workflows are:

- Electrocardiogram (ECG)
- Holter electrocardiography (with heart rate variability (HRV))
- Exercise testing:
  - Exercise test
  - Cardiopulmonary stress testing
  - 6-minute walking test
- Echocardiography (TTE)
- Chest X-ray

All the raw data produced from these medical devices (with the exception of the 6-minute walking test that does not contain raw data and the chest X-ray) contain the ECG.
According to the first results of the inventory performed in WP2, it seems that raw data will not be available for exercise testing, while raw data might be available for some ECG and Holter ECG devices. While echocardiography data will be available in DICOM format as single frame and multi-frame images, no information is known at the moment about chest X-ray devices. If available, digital radiographs adhering to NEMA standards (namely in DICOM format) could be integrated in the platform with no further effort.

5.1.1 ECG and Holter ECG Processing

The ECG processing and especially the resting ECG and Holter ECG processing are of paramount importance in the routine clinical practice environment.
These examinations have many common points and common processing as illustrated in the previous chapters.

The ideal situation would be the capability of automatically extract all the identified measurements and features (see D9) from the ECG using appropriate signal processing. Unfortunately such amount of signal processing requires a huge effort that goes far beyond the initial purpose of the HEARTFAID platform. Thus, a more realistic approach we considered is based on the extraction of some significant features and in the provision of some help to the clinicians for the evaluation of the other required features.
In order to accomplish this task we identified the following processing:

- ECG pre-filtering
- QRS detection
- QRS classification

The ECG pre-filtering will be made in the time domain and will include:

- Reduction of the baseline wandering (low frequency noise)
- Reduction of the high frequency noise
- Noise estimation in each channel (to exclude the noisy channel from the processing)
- Identification of periods with specific rhythms (i.e. in case of a period of ventricular fibrillation/flutter, QRS detection has not to be performed in all that period)

In fact, the usual limited number of channels that are available in the Holter ECG (2 or 3 typically) can make it difficult to properly detect QRS complexes in case of high noise (and high noise can happen in long-term ECGs mainly due to the movement of the patient and to an imperfect contact of the electrodes during the 24 hours). Furthermore, when high noise is present on a single channel its use in the signal processing chain can deteriorate the QRS detection performance introducing false positive and missing some QRS detections (false negative) (Kohler *et al.*, 2002; Fernantez *et al.*, 2003). Thus, noise estimation in each channel can help in dynamically select or exclude some channels from the signal processing chain and thus improving the overall performances of the QRS detection. Furthermore, periods with ventricular fibrillation/flutter have to exclude from the QRS detection because the algorithm could erroneously mark as QRS all the sequence of ventricular flutter waves (Fernantez *et al.*, 2003; Amman *et al.*, 2005; Jekova *et al.*, 2002; Jekova and Krasteva, 2004; Tikannen, 1999).
Once the ECG signal has been pre-filtered, the QRS detection can be performed. Different techniques will be considered even if the spatial velocity seems to be the most reliable method. Spatial velocity is obtained as a derivative filtering of each pre-filtered channel and, in order to increase the signal to noise ratio, all the derivatives of the “good” channels are summed in absolute value. Typically an adaptive threshold is applied for the QRS detection. In order to avoid too much near detections (and to avoid the detection of the T-wave as a QRS), the adaptive threshold is artificially increased after the first crossing. In case of very close detections an algorithm is applied to select among all the points the true QRS. Once the QRS is identified then an algorithm for the detection of the R-peak (fiducial point) (in the neighbourhood of the detected point) is applied. The overall schema is shown in Figure 5.1 (Kohler *et al.*, 2002).

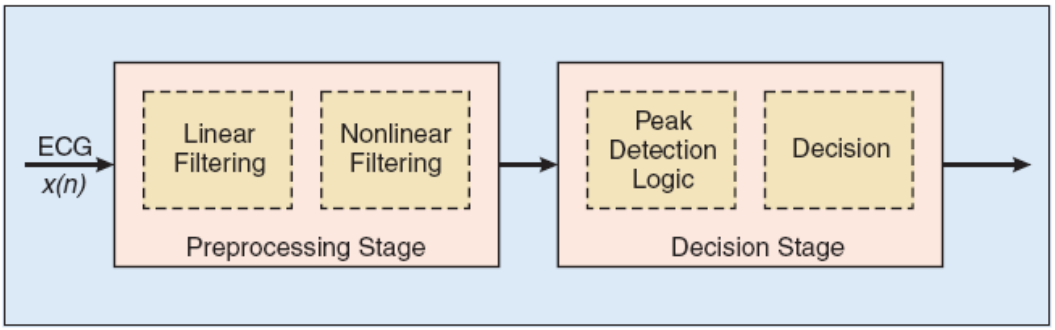


Figure 5.1. *The ECG pre-filtering and QRS detection schema.*

The annotated databases available at the PhysioNet web site will be used for the development and the test of the algorithms that in a first instance will be developed in Matlab.
After QRS detection, it is important to identify the normal and the abnormal beats. This operation requires a morphological analysis of the detected QRSs and a rhythm analysis in order to identify all atrial premature beats (same morphology but premature) (Forbes, 1995). This operation is required for a proper selection of all normal beats. The QRS classification is performed in the time-domain with cross-correlation algorithms (beat alignment can be checked again and the fiducial points adjusted) or with clustering algorithms. In this last case, assuming that each QRS complex can be represented by a multidimensional vector, the classification problem can be described as finding the partitioning of the high-dimensional parameter space in such a way that each subspace contains vectors of QRS complexes of the same morphology (Talmon, 1983).
This problem consists of two subproblems, which influence each other:

- Which parameters are the best descriptors of the morphology of the QRS complex
- Which clustering or partitioning algorithm is most suitable given the parameter set

5.1.2 Processing of Echocardiography Images

Echocardiography is the main imaging technique used in the diagnosis and follow up of heart failure. It is a complex examination, involving different kinds of images and signals, useful to derive quantitative parameters of chambers size and shape, diastolic and systolic functions and for semi-quantitative assessment of valvular functions. As for the ECG, the ideal situation would be to automatically extract all such parameters. However this task is far beyond the scopes of HEARTFAID platform and, overall, its feasibility is not clear, since it would require at least adherence to a complex acquisition protocol. So, it is reasonable to restrict to some specific parameters, where image processing could turn out more useful. Other parameters, edited by the sonographer, would eventually enter the platform via HL7 messages (supported by some of the devices available in HEARTFAID validation sites) or by other means to be considered in other WPs.

In Section 2.3, we showed how manual border delineation is prone to intra- and inter-observer variability, with dramatic consequences in the computation of derived parameters, like ejection fraction.
Thus, the development of an automatic procedure for 2D echocardiographic image segmentation would turn out extremely useful, since it would enable immediately the determination of the classical parameters for chambers quantification suggested by the American Society of Echocardiography.
From the general segmentation scheme for US images described in Section 0, segmentation will be performed on multi-frame image sequences, exploiting spatio-temporal coherence and a shape prior obtained through learning. The apical 2C/4C views and short axis parasternal views will be considered at least, given their prominent role in chambers quantification.
The whole procedure includes:

1. Image filtering. Anisotropic diffusion filters for image smoothing and noise reduction will be evaluated. Methods for US intensity distribution correction via non-linear filtering will also be analyzed.
2. Automatic procedures for cardiac cavities localizations.
3. Rough segmentation of the cavities. This will be need to provide good initialization for segmentation refinement and will be based on a mimetic strategy.
4. Active contours models for consistent segmentation of cardiac structures from image sequences.
5. Design of suitable shape priors for segmentation reinforcement and their implementation in the chosen active contour scheme.
6. Pattern recognition methods for the identification of valvular leaflets, essential to separate atria from ventricula in volume computations
7. Algorithms for automatic detection of the cycle phases (especially end-systole and end-diastole)
8. Feature extraction by linear measurements and standardized formulas for chambers quantification.

To the best of our knowledge, no method in the literature met good performance, automation, robustness and adaptability. So, any serious attempt to automatic border delineation to be used successfully in a platform should allow some user interaction in order to cope with algorithm failure or small imperfections. Such issues will be considered carefully. It is accepted as a general rule in the medical imaging community that if a computer based method provides reasonable measurements for most patients but fails dramatically for other patients with poor acoustic windows, unusual pathology or prostheses, most physicians will tend to be forgiving. However, an algorithm that works only for high-quality images will not meet any requirements of the health care operators.

Furthermore, the possibility to fuse the segmented 2D views of anatomical structures into the 3D space will be explored, analyzing whether a sufficient precision is achievable. The challenging problem of automatic view understanding (i.e. for example automatically distinguish a parasternal view from an apical one) will also be considered.

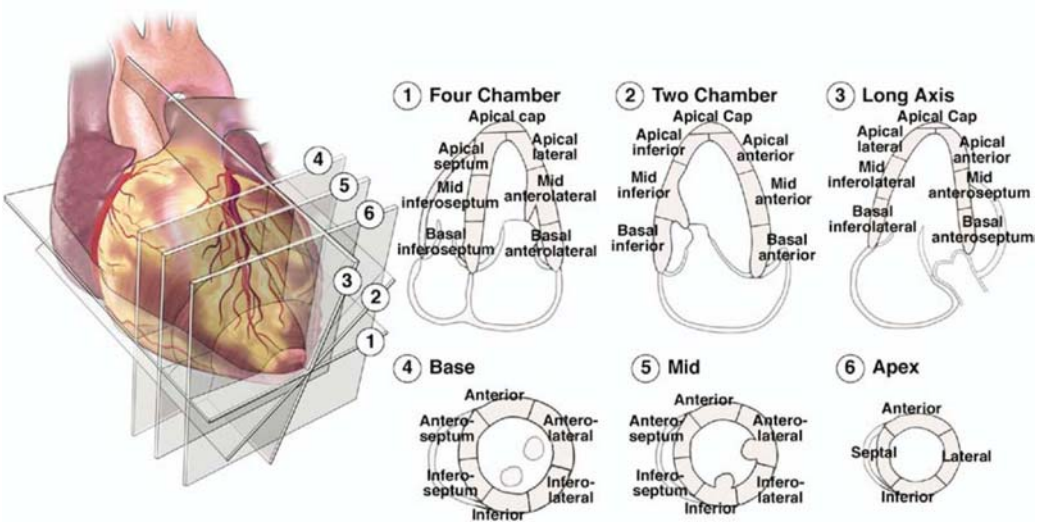


Figure 5.2. *The 17 segments model of the left ventricle (after ASE Guidelines by Lang et al. (2005)). Disregarding the apical cap one obtains the more commonly used 16 segments model*

As discussed again in Section 2.3, the method for segmental scoring of wall motion and thickening, although being sufficient for routine use, is semi-quantitative and could be ameliorated by use of automatic methods. The aim is to refine and make quantitative measures to be eventually translated into the standardized scoring system, in order to provide results easy interpretable by health care operators in the routine clinical environment.

See Figure 5.2 for the nomenclature of standardized segments, Figure 5.3 for the usual graphical representation used in clinical practice and Figure 5.4 for a real example of reduced wall motion.

The feasibility of this task depends tightly from a good segmentation performance and the ability to incorporate different echocardiographic views. A model-based approach has to be used, similar to those described in Section 0 below.

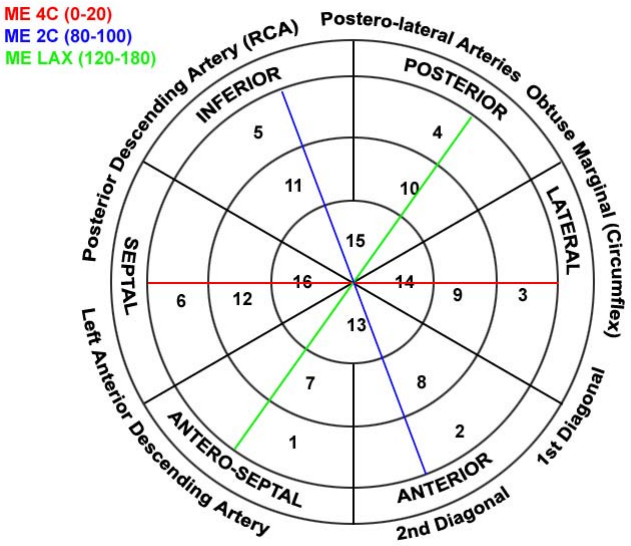


Figure 5.3. *The scoring of segmental wall motion is usually performed filling each sector in the above diagram with a number ranging from 1 (normal motion) to 5 (aneurysmal). Here 16 segments are considered, the apical cap being excluded*

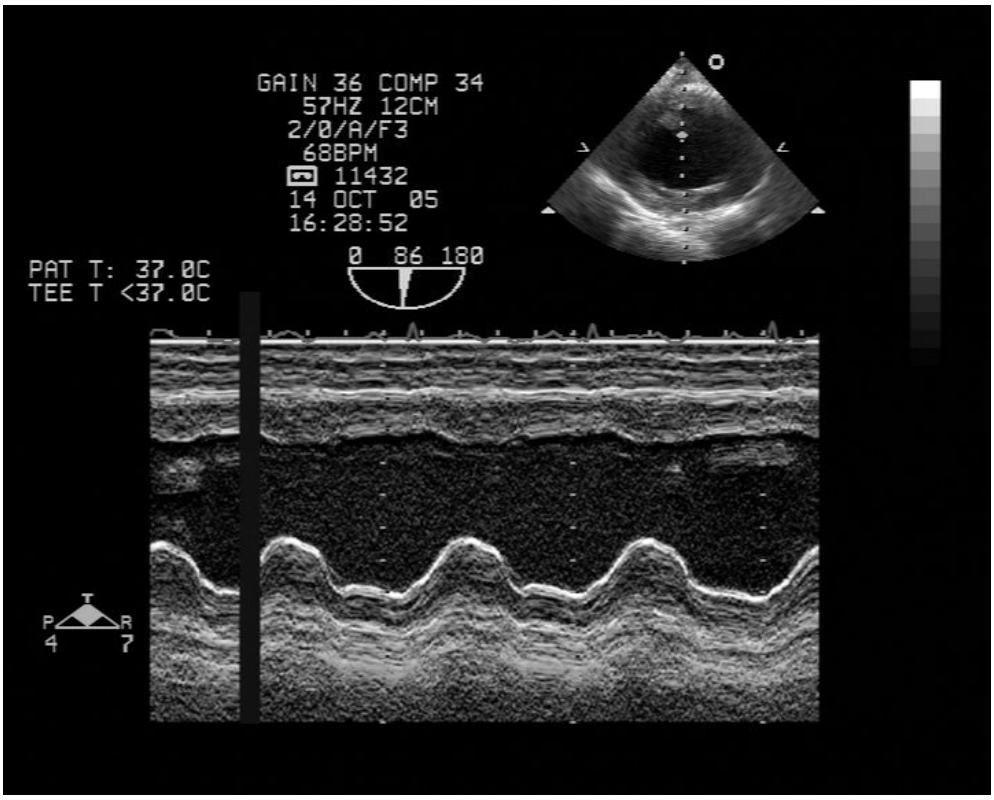


Figure 5.4. *Hypokinesia of the inferior assessed visually from a 2D-targeted M-mode view. The upper cavity border in the image does not exhibit the undulatory movement visible in the lower border*

5.1.3 List of Representation Features for the clinical routine practice

Below we critically review from a signal processing viewpoint the features extracted by the clinicians relevant for HF evaluation.

ECG

From the resting ECG several features are usually extracted by the clinicians to be used in the HF evaluation (Macfarlane and Lawrie, 1989). The typical features, as stated by the clinicians, are:

Sinus rhythm Y/N

Heart rate …. beats/min
PR interval …. mm
QRS duration …. mm
QT interval …. mm

Presence of abnormalities Y/N

Atrial flutter Y/N
Atrial fibrillation Y/N
(Atrial flutter and atrial fibrillation are better detected, especially in case of paroxysmal episodes, with a 24-hour Holter ECG)

Atrio-ventricular conduction abnormalities Y/N
- *grade I A-V block Y/N*
- *grade II A-V block type I Y/N*
- *grade II A-V block type IIY/N*
- *grade III A-V block Y/N*

Intra-ventricular conduction abnormalities Y/N
- *Left bundle branch block Y/N*
- *Right bundle branch block Y/N*

ECG signs of ischemia Y/N
Describe possible reasons (text):
- *i.e. ST depression 3 mm in leads V3 – V6.*
- *i.e. q wave in inferior leads*

Left ventricular hypertrophy: Y/N
(criteria: Sokolow-Lyon S1-S2 + R5-6).

Nowadays several ECG device manufacturers produce “interpretive resting ECG devices” that are able to automatically calculate the main measurements on the ECG signal and a diagnostic proposal.

Thus, commonly all signal processing is performed in the medical devices itself and then the ECG signal, with the result of the processing (measurements and diagnostic proposal) can be transferred to a host computer either to a generic third-party software that implements the implements the communication protocol and knows the data format used by the medical device software for the storage of the examinations or to a software produced by the medical device manufacturer itself.
In all this procedure there is a major problem. Even if standards (i.e. SCP-ECG or HL7 aECG (Brown *et al.*, 2007) are made available from the international Standardization Development Organizations (SDO), several manufacturers do not implement or do not properly implement these standards, thus practically hampering the possibility of developing software able to interface their medical devices and to properly read the acquired examinations (see Reynolds *et al.*, 2004).
Apart from this main problem, in case raw data of a resting ECG will be acquired from an ECG device, we will have typically two possibilities:

1) The examination is exported in form of a textual final report (more common situation). In this case, all the necessary data are already available for their insertion in the clinicians form used for the Heart Failure evaluation.
2) The examination is exported as raw data with some additional information (sampling rate, LSB, etc.) necessary for properly reconstructing the signal. In this case the typical routine processing described in Section 5.1 could be applied.

Holter ECG

From the Holter ECG several features are usually extracted by the clinicians to be used in the Heart Failure evaluation. The typical features, as stated by the clinicians, are:

Sinus rhythm Y/N

- *if N specify …. (text)*

24 heart rate:

- *mean …. numeric beats/min*
- *minimum …. numeric beats/min*
- *maximum …. numeric beats/min*

daytime heart rate:

- *mean …. numeric beats/min*
- *minimum …. numeric beats/min*
- *maximum …. numeric beats/min*

nighttime heart rate:

- *mean …. numeric beats/min*

- *minimum …. numeric beats/min*
- *maximum …. numeric beats/min*

Maximum R-R …. numeric seconds

Isolated Ventricular ectopic beats …. numeric N/24 h

Ventricular couples …. numeric N/24 h

3 or more Ventricular ectopic beats …. numeric N/24 h

- *maximum N ….. numeric*
- *maximum heart rate … numeric Beats/min*

Isolated Supraventricular ectopic beats …. numeric N/24 h

Supraventricular couples …. numeric N/24 h

3 or more supraventricular ectopic beats …. numeric N/24 h

- *maximum N ….. numeric*
- *maximum heart rate … numeric Beats/min*

Other arrhythmias …. Text

Atrio-ventricular conduction abnormalities Y/N

- *grade I A-V block Y/N*
- *grade II A-V block type I Y/N*
- *grade II A-V block type IIY/N*
- *grade III A-V block Y/N*

Intra-ventricular conduction abnormalities Y/N

- *Left bundle branch block Y/N*
- *Right bundle branch block Y/N*

Signs of ischemia Y/N describe (text):

Heart rate variability (HRV):

- *SDNN ms*
- *SDANN ms*
- *rMSSD ms*
- *pNN50 %*
- *Total power ….*
- *HF ….*
- *LF ….*
- *VLF ….*

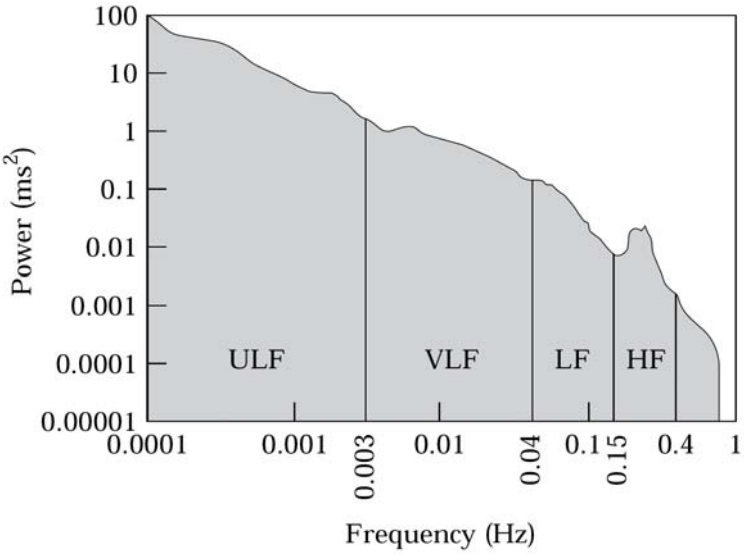


Figure 5.5. *Estimated power spectrum density from a 24h Holter ECG. The sectors relative to HF, LV, VLF are used as representation features (ESC-NASPE, 1996).*

All the acquisitions performed by Holter ECG data logger are reviewed by specialized personnel on the Holter review station, where the analysis performed by the reviewing software is confirmed or corrected by the clinicians. The software usually allows the production of a final report that typically contains all the features/measurements listed above.

Thus, commonly all signal processing is performed in the reviewing station itself and then the ECG signal, with the result (or a subset of it) of the processing (measurements, features and results of the HRV analysis) can usually be exported on an external file for its transfer to a third-party system (Zareba *et al.*, 1998)(Kemp *et al.*, 1992).

Of course, the third-party software has to know the data format used by the review station for the exporting of the examinations.

In all this procedure there is a major problem. Even if several disclosed formats (i.e. MIT/BIH, ISHNE (uncompressed), EDF or EDF+) are made available from the international scientific community, several manufacturers do not implement these formats or do not properly implement these formats, thus practically hampering the possibility of developing software able to interface their medical devices and to properly read the exported examinations.

Apart from this main problem, in case raw data of a Holter ECG will be acquired from a Holter review station, we will have typically two possibilities:

1. The examination is exported in form of a textual final report (more common situation). In this case, all the necessary data are already available for their insertion in the clinicians form used for the Heart Failure evaluation.
2. The examination is exported as raw data with some additional information (sampling rate, LSB, etc.) necessary for properly

reconstructing the signal. In this case the typical routine processing described in Section 5.1 could be applied.

Actually, whereas arrhythmias are easily identified by the clinicians at the reviewing, the N-N intervals are of basic importance for the HRV analysis and this processing can save a significant amount of work from the clinicians and help also in the discovery or the assessment of new potential features in HRV.

STX

From the STX several features are usually extracted by the clinicians to be used in the Heart Failure evaluation. The typical features, as stated by the clinicians, are:

Basal heart rate	*…. numeric beats/min*
Maximum predicted heart rate	*220-age in years…. numeric beats/min*
Maximum heart rate	*…. numeric beats/min*
% of predicted heart rate	*…. numeric %*
Basal systolic/diastolic blood pressure	*…./….. numeric mmHg*
Maximum systolic/diastolic blood pressure	*…./….. numeric mmHg*

Protocol

Maximum load	*numeric …. Watts* *…… METS*
Total minutes of exercise	*numeric …. min*
Abnormalities in baseline ECG	*Y/N describe …. text*
Criteria for test interruption	*….. text*
Test was positive for ischemia	*Y/N describe ---- text*
Other abnormalities	*….. text*

The usual way of working of the cardiopulmonary exercise testing systems is that at the end of the test, the medical device allows the specialized personnel for the revision and the editing of the final report. The software usually allows the production of a final report that typically contains all the features/measurements listed above.
Thus, commonly all signal processing is performed in the medical device itself and then the raw data, with the result (or a subset of it) of the processing (measurements, features and results) may be exported on an external file for its transfer to a third-party system.

Of course, the third-party software has to know the data format used by the review station for the exporting of the examinations and unfortunately no specific standards are available for this kind of examination. A more generic suitable standard might be the EDF format (Kemp *et al.*, 1992), but it is not largely adopted by manufacturers of these medical devices. Thus, specific software has to be implemented for each brand/model system. For this reason, we can exclude the possibility to have raw data available for this kind of medical device and we will consider as best effort the automatic or manual (via web form) import of a textual final report containing all relevant information for the HF domain.

CPX

From the cardiopulmonary exercise testing several features are usually extracted by the clinicians to be used in the Heart Failure evaluation. The typical features, as stated by the clinicians, are:

Basal heart rate …. *numeric beats/min*
Maximum predicted heart rate *220-age in years….numeric beats/min*

Maximum heart rate …. *numeric beats/min*
% of predicted heart rate …. *numeric %*

Basal systolic/diastolic blood pressure …./….. *numeric mmHg*
Maximum systolic/diastolic blood pressure …./….. *numeric mmHg*
Protocol
Maximum load *numeric …. Watts*
numeric …. METS
Total minutes of exercise *numeric …. min*

Abnormalities in baseline ECG *Y/N describe …. text*

Criteria for test interruption ….. *text*

Test was positive for ischemia *Y/N describe ---- text*

Other abnormalities ….. *text*

Peak VO2/Kg (peak oxygen uptake/Kg) …. *numeric ml/Kg/min*
Peak VCO2/Kg …. *numeric ml/Kg/min*
VO2 AT (oxygen uptake at the anaerobic threshold) …. *numeric ml/Kg/min*
Basal RQ (respiratory quotient) …. *numeric*
O2 pulse (ml O2/beat) … *numeric ml/beat*
Basal saturation SO2 …. *%*
Peak SO2 …. *%*
Ve/VCO2
Ve/VCO2 (ventilation/VCO2) …. *numeric*

The usual way of working of the cardiopulmonary exercise testing systems is that at the end of the test, the medical device allows the specialized personnel for the revision and the editing of the final report. The software usually allows the production of a final report that typically contains all the features/measurements listed above.
Thus, commonly all signal processing is performed in the medical device itself and then the raw data, with the result (or a subset of it) of the processing (measurements, features and results) may be exported on an external file for its transfer to a third-party system.
Of course, the third-party software has to know the data format used by the review station for the exporting of the examinations and unfortunately no specific standards are available for this kind of examination. A more generic suitable standard might be the EDF format (Kemp *et al.*, 1992), but it is not largely adopted by manufacturers of these medical devices. Thus, specific software has to be implemented for each brand/model system.
For this reason, we can exclude the possibility to have raw data available for this kind of medical device and we will consider as best effort the automatic or manual (via web form) import of a textual final report containing all relevant information for the HF domain.

6MWT

From the 6MWT, the typical features used in the HF evaluation, as stated by the clinicians, are:

	Baseline	End
BP (mmHg)	……	……
HR (bpm)	……	……
SpO_2 (%)	……	……
walking distance (m)	……	

No raw data are associated with this examination. Specific web forms will be filled in with the collected measurements, thus a sort of final textual report will be available for each 6MWT.

Echocardiography

A great number of representation features can be extracted from TTE, regarding chambers size and shape, systolic and diastolic function, valves, Aorta and inferior vena cava. The list below is organized by considering the anatomical structure on which the examination is performed.

Left ventricle

For the quantification of the left ventricle the medical partners suggested to consider the following features:

end-diastolic diameter	*….. (numeric) mm*
end-systolic diameter	*….. (numeric) mm*
interventricular septum diastolic thickness	*….. (numeric) mm*
posterior wall diastolic thickness	*….. (numeric) mm*
end-diastolic volume	*….. (numeric) ml*
end-systolic volume	*….. (numeric) ml*
ejection fraction	*….. (numeric) %*
Method :	*Simpson's 2C, 4C, biplane*

All the features can be derived from linear measurement on 2D echo or M-mode images. However, the use of linear measurements for quantification of volumes (such as Teichholz method) is not recommended by ASE, since it requires too strong geometric assumptions about the shape of the LV and it is extremely sensitive to incorrect view selection. Simpson's method (and in particular its biplane variant, known as modified Simpson's rule) is the preferred method for assessing end –diastolic and –systolic volumes and, in turn, the ejection fraction. This method requires border delineation, which, as already said, is prone to intra- and inter- observer variability. See Figure 5.6 for an application of Simpson's rule.

Right ventricle

For the quantification of the right ventricle, the following parameters have been individuated:

end-diastolic diameter	*….. (numeric) mm*
TAPSE	*….. (numeric) mm*

The first can be obtained by linear measurement from the apical 4C view and refer to chamber size. TAPSE is instead a parameter of RV systolic function and it is based on the quantification of the displacement of the tricuspid annulus plane from M-mode echocardiogram.

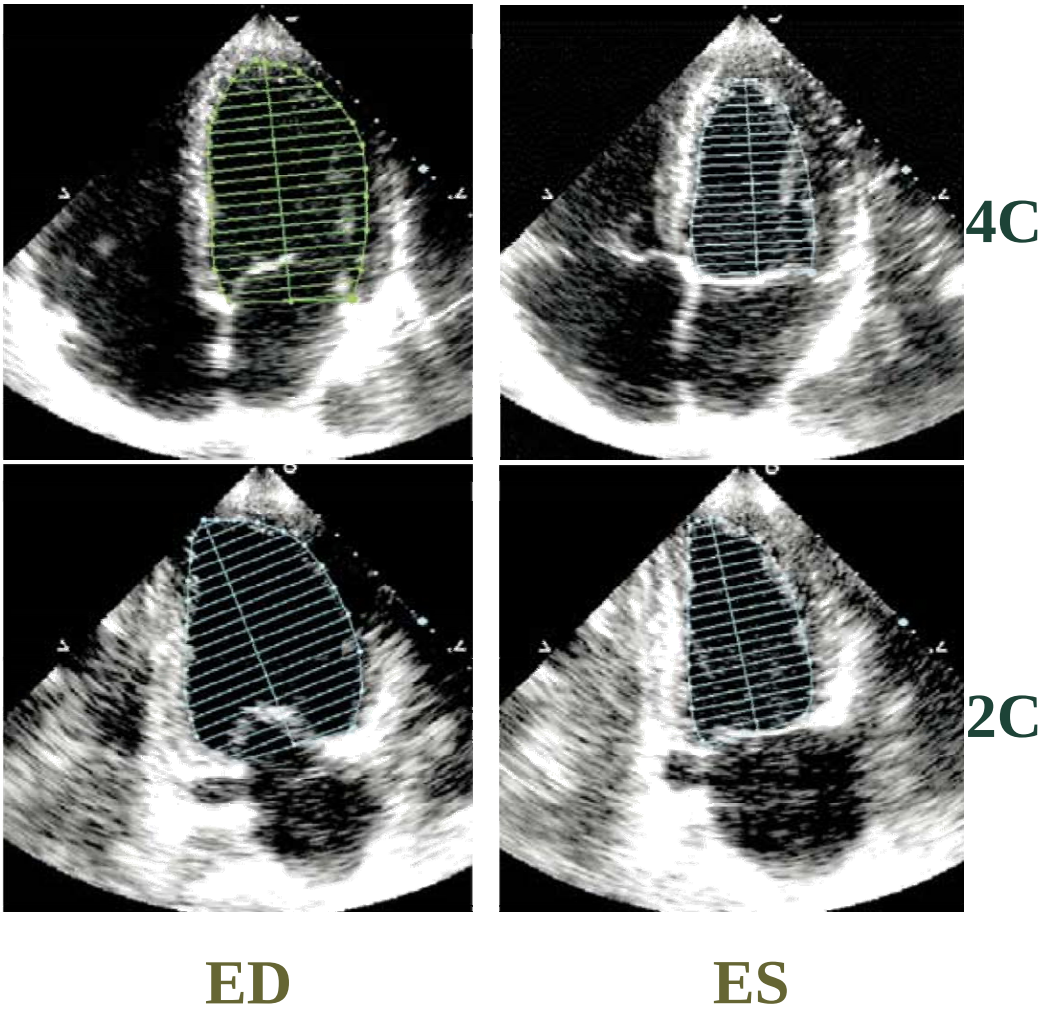


Figure 5.6. *Computation of LV EF by biplane Simpson's method. The volume of the left ventricle cavity is estimated at end diastole (ED) and end systole (ES) by fusing the four and two chambers apical views (4C/2C respectively). The LV ventricle is approximated by a stack of elliptical cylinders whose semi-radii are the radii extracted from the two views*

Left atrium

Also for the quantification of the left atrium only a linear measurement of the chamber has been selected, namely the

Anteroposterior diameter ….. *(numeric) mm*

LA enlargement is actually a marker of both the severity chronicity of diastolic disfunction. The measure is usually performed from M-mode images 2D-targeted via a parasternal short axis image at the level of the aortic valve. However, LA linear measurements can be accompanied by area and volume measurements, in

order to get a better characterization. See Figure 5.7 for an example of LA area computation.

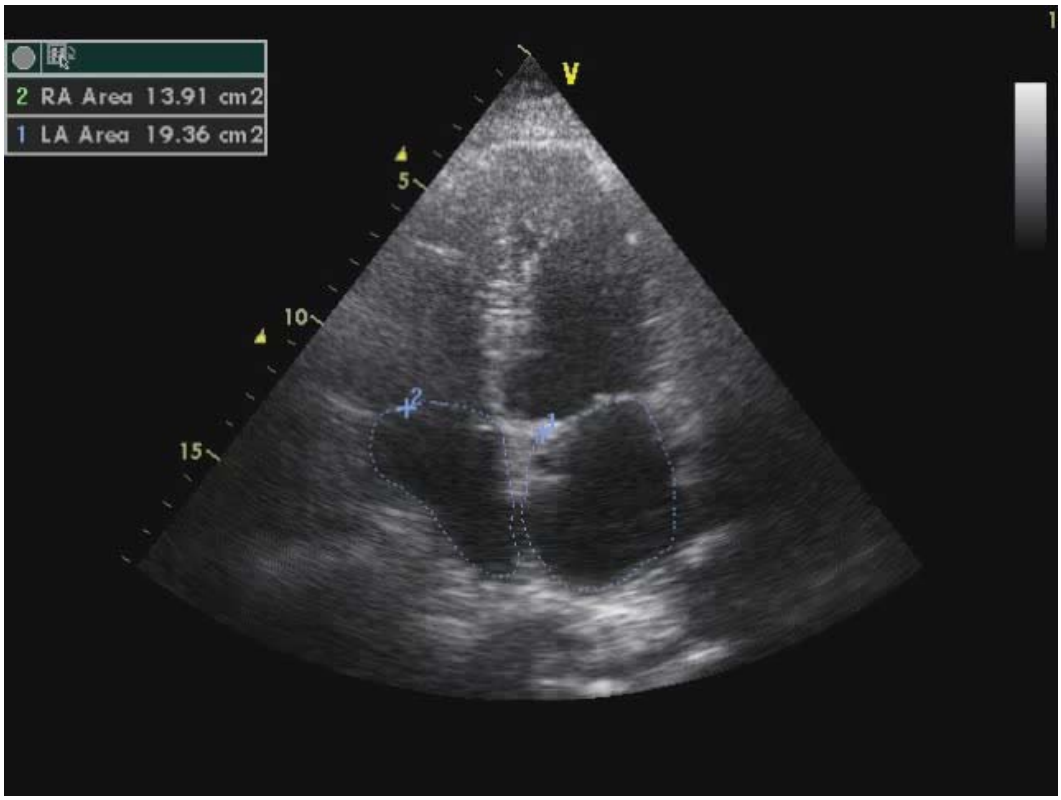


Figure 5.7. *Border delineation of the atria in a 4C apical view. The computed areas are shown in the upper left box*

Aorta

Two features are considered for the quantification of the Aorta, namely

Root diameter	*….. (numeric) mm*
Ascending aorta diameter	*….. (numeric) mm*

They both consist in linear measurements, usually performed from a long azis parasternal view. See Figure 5. for an example of measurement using M-mode.

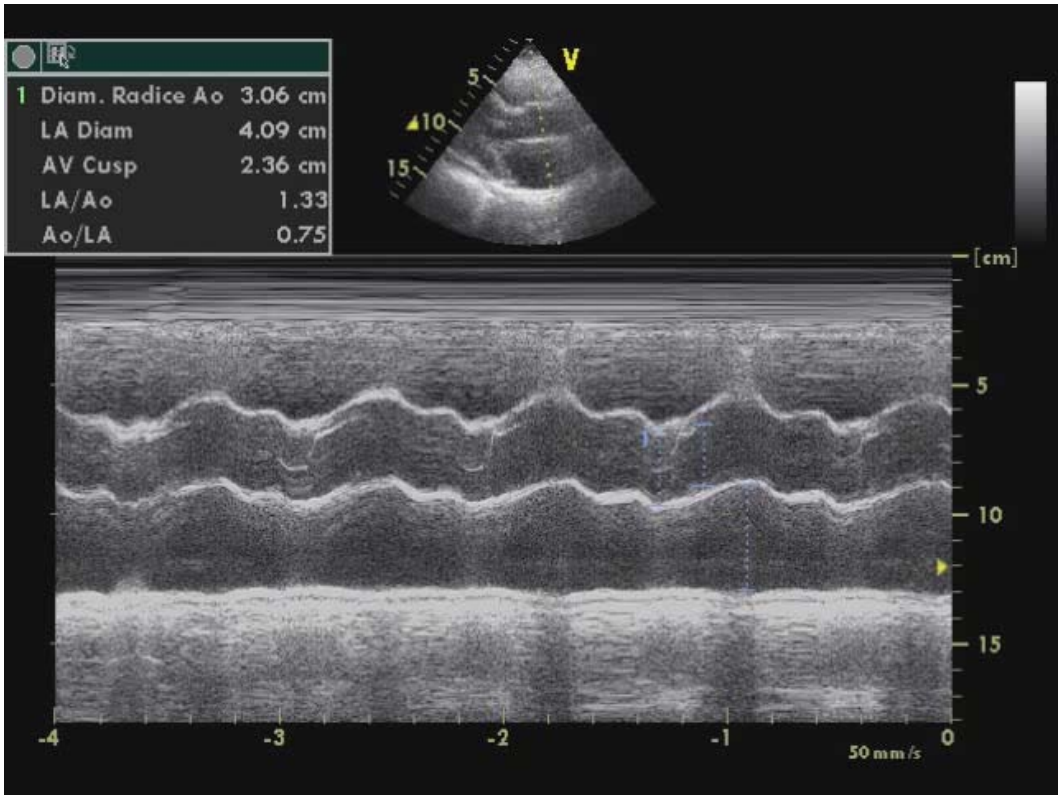


Figure 5.8. *Assessment of root Aortic diameter ("Diam. Radice Ao" in the picture) by 2D-targeted M-mode*

Aortic Valve

TTE provides semi-quantitative methods for the assessment of aortic valve regurgitation and stenosis. The following parameters are obtained by Doppler techniques.

Aortic regurgitation	*Y/N*	*grade: +,++,+++,++++*
Aortic stenosis	*Y/N*	*gradient mean….. (numeric) mmHg*
		gradient peak ….. (numeric) mmHg

Mitral valve

The examination of the mitral valve permits to understand not only the valvular function itself but gives much insight in the assessment of diastolic function.

E max	*….. (numeric) m/s*
A max	*….. (numeric) m/s*
E/A ratio	*….. (numeric)*
Deceleration time	*….. (numeric) m/s*

Mitral regurgitation	*Y/N*	*grade: +,++,+++,++++*
Mitral stenosis	*Y/N*	*gradient mean….. (numeric) mmHg*
		gradient peak ….. (numeric) mmHg
Diastolic dysfunction		*Y/N*
(EROA, Doppler shift)		*……*

The first four parameters are obtained by Doppler techniques and are indicative of the blood flow from the left atrium to the left ventricle. Since they represent the filling pattern of the ventricle, they are markers of diastolic function. Usually one may distinguish among three kinds of filling pattern (see Figure 5.9 for a scheme and Figure 5. for a real example). Actually a fourth pattern, called pseudonormalized filling, has been described and refers to moderate diastolic dysfunction.

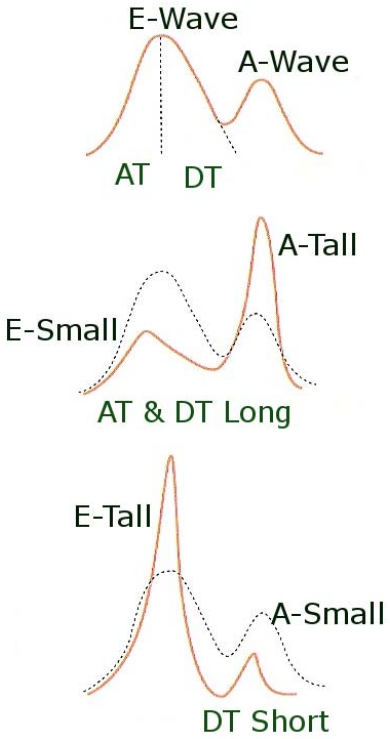


Figure 5.9. *Filling pattern of the left ventricle. Schematic representation of the blood flow through the mitral valve (see the next Figure for a real example). From top: Normal, Slow Relaxation pattern and Restrictive pattern*

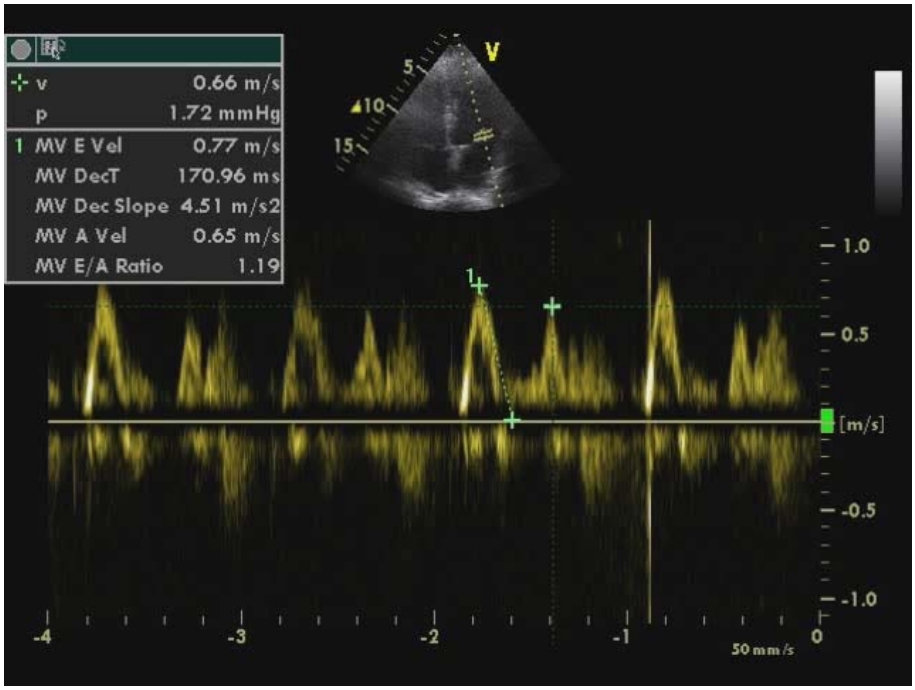


Figure 5.10. *Doppler assessment of mitral flow*

By conventional Doppler imaging, pseudonormalized filling is practically indistinguishable from the normal filling pattern. It can be however discriminated by the relative new modality called Tissue Doppler Imaging, applied to the mitral annulus (such modality is available in some of HEARTFAID validation sites). A reduction of the analogous of the peak early diastolic mitral flow velocity *E* (that is the so-called peak early diastolic mitral annulus velocity *E*') is indicative of pseudonormalized filling. For this reason, when available, it would be reasonable to consider this parameter as well. For the quantification of mitral regurgitation and stenosis there exist many techniques, for which we refer to the American Society of Echocardiography specific guidelines (Zohgbi *et al.*, 2003).

Tricuspid valve

Parameters of regurgitation and indirect estimation of pulmonary artery pressures are considered for the tricuspid valve. Notably, the latter estimation requires as well the evaluation of the inferior vena cava diameter and of its collapsibility index from the subcostal view.

Tricuspid regurgitation	*Y/N grade: +,++,+++,++++*
Pulmonary artery pressure	*….. (numeric) mmHg*

Contractility

As contractility features, the global segmental scoring discussed at the end of Section 0 has been selected by the medical partners.

Chest X-ray

In D9 few representation features extracted from the chest X-ray have been described:

Cardiothoracic ratio ….. (numeric)		
Pulmonary circulation congestion	*Y/N*	*describe (text) …*
Pulmonary oedema	*Y/N*	*describe (text)….*
Effusion	*Y/N*	*describe (text) ….*

The only numeric value is given by cardiothoracic ratio, which is obtained by simple linear measurements and is indicative of cardiomegaly. At the moment no processing activity is planned for this modality; however segmentation and categorization of digital chest radiographs, including quantifications of pleural effusion, would be considered in the future, according to eventual suggestions by the medical partners.

5.2 Long-Term Research Environment

5.2.1 Signal Processing

In the long-term research environment we have to include also all the examinations that are only part of non-routine workflows. These clinical examinations are:

- Spirometry
- Bioimpedance
- Autonomic parameters other than HRV:
  - Blood pressure variability
  - Baroreflex sensitivity
  - Cardiac autonomic modulation parameters
- Additional devices for specific measurements (only for research workflows)

For this kind of examinations it is usually even more difficult to have the raw data available and, according to the first results of the inventory performed in WP2, it seems that raw data might be available only from ABP devices. Very unlikely the other devices will allow the transfer of the raw data to the HEARTFAID platform.
Thus, the only areas of intervention for signal processing are in the ABP devices (blood pressure variability algorithms), advanced ECG analysis and the use of additional devices for specific measurements.
In the field of BPV the algorithms will be devoted to artefact rejection and to the evaluation at least of the features identified in D9. Of course the possibility of identifying other additional features that can be evaluated starting from the raw data will be taken into account.
In the field of ECG and mainly of HRV, new innovative algorithms might be considered. In fact, most physiological signals, including heart rate, are non-stationary and nonlinear in nature. This is clearly the case for the autonomically-regulated cardiac rhythm. Thus, frequency analysis and even STFT may fail to reveal ‘instantaneous’ time localised information contained within the R-R time series. Moreover, assuming the nonlinear nature of the signal, turn all available linear data analysis techniques inadequate on their own to correctly assess physio-pathological ‘state’ of the system. A combination of linear and non-linear HRV analysis techniques on an automated system, which evaluates both the linear and nonlinear trends of the signal, might be of great help. However, the computationally intensive algorithms for these non-stationary and nonlinear analyses, conventionally implemented offline have precluded the use of these powerful analytic tools in real-time. HF platform however might offer the possibility of simultaneously quantifying the time-domain, the non-stationary and the non-linear dynamic nature of inter-beat heart rate fluctuations with reference to the sympathetic and parasympathetic inputs in live or pre-recorded signal segments significantly contributing in the field of HRV analysis.

Finally, the use of additional devices for specific measurements will be considered after the first results produced in the validation sites. Clinicians will evaluate if the first outcomes will suggest the use of some new device and in that case the selection of the device and its integration in the platform will be analysed in order to realize best effort integration in the attempt of making the raw data also available. If raw data will be available, then the signal processing that can be applied will be evaluated at that moment.

Below we review in detail, from a signal processing viewpoint, the features extracted by the clinicians relevant for research in the HF domain.

HRV parameters from Holter ECG

- *PIsd*
- *HRsd*
- *VLFpi*
- *LFpi*
- *HFpi*

The same considerations exposed for the clinical routine practice apply.

SPT

The important information evaluated from the raw data for the HF domain is:

- Forced vital capacity (FVC)
- Forced expiratory volume (FEV)
- Peak expiratory flow (PEF)

The most common case is with no raw data available. In this case the above parameters evaluated by the software running on the medical device itself will be read by the operator and a specific web forms will be filled in with these collected measurements, thus a sort of final textual report will be available for each spirometry test. If raw data will be available, then the signal processing that can be applied will be evaluated at that moment.

BIA

The typical parameters extracted from the measured values are:

- Total body water (TBW)
- Body cellular mass (BCM)
- Intracellular water (ICW)
- Fat Mass (FM)
- Extracellular water (ECW)
- Phase angle (PA)
- Muscular mass (MM)
- Sodium/potassium exchange ratio (Nae/Ke)
- Basal metabolism (BMR)

As well known, total boby water is the most relevant parameter in HF.
The lack of standards for data format related to this modality makes it very difficult to have an intelligible export of the acquired examinations containing raw data (in case of PC-based device). Either no information is available or only a textual final report will be available. In the first case a web form will be used to insert the TBW value for the acquired examination, while in the second case an automatic reading of the TBW value will probably be feasible with the automatic insertion of the read value in the HEARTFAID platform. If raw data will be available, then the signal processing that can be applied will be evaluated at that moment.

BRV

The typical features extracted from the measurements (once eliminated the artefacts) are the variability indexes expressed as the standard deviation of the measurements collected during the whole observation period.
The overall parameters considered for this modality are:

- SBPsd (standard deviation)
- DBPsd
- MAPsd
- VLFsbp
- LFsbp
- HFsbp
- VLFdbp
- LFdbp
- HFdbp
- VLFmap
- LFmap
- HFmap

These values are obtained in the time-domain (the standard deviations) and in the frequency-domain (the very low frequency, low frequency and high frequency peak values).

BRS

The overall parameters considered for BRS are:

- alpha coefficient (AlfaLF, AlfaHF)
- transfer function (H-LF, H-HF)
- sequence technique:
    - slope Seq++
    - slope Seq--
    - baroreflex effectiveness index (BEI).

It seems that the devices currently available in HEARTFAID validation sites will not be able to export raw data. If raw data will be available, then the signal processing that can be applied will be evaluated at that moment.

CAM

Other cardiac autonomic modulation parameters that will be collected are:

- SBPm (media or average)
- DBPm
- MAPm
- pulse interval (PI)m
- HRm

These numeric values are obtained by Holter devices. The same considerations apply.

5.2.2 Perspectives in Cardiac modelling

Cardiac modelling refers to the process of building an abstract representation of the heart that could be used for cardiac image analysis. The two main-goals of cardiac modelling described in the literature are:

1. Incorporate a priori shape knowledge in the segmentation process
2. Extract parameters of cardiac shape and function

Actually, as we have discussed, inclusion of a priori information is essential in dealing with segmentation of cardiac structures; so model-driven methods are welcome since they can translate the segmentation problem into a model fitting problem (i.e. instantiation of the abstract model into the patient's anatomy depicted in the image data). Considering the parameters of the instantiated model, by the same token, one can derive dozens of cardiac shape and functional attributes of different classes ranging from global to regional, classical or exotic ones. Further it would be useful to fuse in an unified picture heart functional attributes (regarding motion, perfusion, metabolism) gathered from a variety of imaging modalities.

The amount of the literature regarding cardiac modelling has grown impressively in the last years. A good but dated critical review is presented in Frangi *et al.* (2001), featuring more than 200 bibliographic items, to which we refer for a glimpse on this topic. Still, cardiac modelling is an active research field where many issues remain to be solved. Actually in spite of the large number of attempts, no approach seems to have simultaneously achieved robustness, automation, model flexibility and computational speed. In clinical environments, also when dealing with high-end imaging techniques like MRI, contours are still manually delineated.

In general, the trade-off between model flexibility to get realistic shapes and constraints to achieve robustness should be addressed more carefully in the future. In particular, one may ask for a faithful and realistic representation that should be compact (or minimal in some sense), possibly learned from a training set of examples. In the same vein, compact representation may boil down the intensive computations involved in the problem.

Further, cardiac modelling could provide many shape and motion parameters that deviates from classical ones. However in the literature, the clinical evaluation of these new parameters has been extremely limited. In this setting, HFP could represent a good place to start evaluation of non-traditional descriptors, for testing and benchmarking purposes.

Finally, almost every model available in the literature is based only or geometry or *virtual physics* (mass and springs models, moving interfaces, thin membrane and the like) methods. It would be interesting to model the heart from a biomechanical and electromechanical point of view (for the description of some attempts in this direction, see Sermesant *et al.*, 2003).

5.2.3 Quantification of LV Dyssynchrony

According to ACC/AHA guidelines (Hunt *et al.*, 2005; see also the medical bibliography therein), approximately one third of patients with low EF and NYHA class III or IV symptoms of HF manifest a QRS duration >120 ms. This ECG representation of conduction defects is used to identify patients with dyssynchronus ventricular contraction. Although this ECG-derived parameter is poorly correlated to the outcome of possible cardiac resynchronization therapy (CRT) by biventricular pacing, there is no other consensus on the definition of ventricular dyssynchrony. Yet, several global and segmental markers obtained by TTE and MRI are promising.
It has been demonstrated in a randomized trial that CRT can reduce the risk of all cause of hospitalization and death by approximately 20% and, in conjunction with implantable cardioverter-defibrillator, all causes of death by 36% (Cleland *et al.*, 2005), when compared to optimal medical treatment alone. Quantification of LV dysynchrony and prevision of the outcome of CRT could therefore drastically improve the management of a considerable number of HF patients having intraventricular conduction disturbances.

As claimed before, there has been in the last years an increasing interest in deriving more advanced methods for the definition and quantification of LV dyssynchrony, than those relying on QRS duration.
Most echocardiographic approaches are based on TDI measures of longitudinal velocity, which are however biased by passive motion and, thus may not describe accurately cardiac mechanics (see e.g. Yu *et al.* (2004) ; Gorcsan *et al.* (2004), Bax *et al.* (2004), Penicka *et al.* (2004), Kanzacki et al (2004)). Some more recent ones are based on Tissue Doppler Strain imaging (see Dohi *et al.* (2006)), that may be used to get angle-corrected radial strain images of the ventricle.

Also the computer vision community has been engaged with the problem of dyssynchrony quantification. Actually, by using suitable cardiac modelling, dyssynchrony detection and categorization may be formulated as a statistical pattern recognition problem.

In particular, the activation pattern of LV segments may be encoded in a spatio-temporal parametrization of the LV shape, suitable for classification by AI methods to be conveyed by CDSS. It is likely that a suitable parametrization may be drawn directly by image sequences displaying the morphology of the heart (e.g. 2D echo, MRI, ultrafast CT). Some attempts are described in Gotardo *et al.* (2006), where the authors deal with short axis MRI sequences. To deal with US, the main problem to be addressed is accurate segmentation of the LV, as described earlier.

6. Requirements of the HEARTFAID Clinical Decision Support System

The first step in designing and developing the HEARTFAID CDSS[2] consists in identifying its overall *requirements*, on the basis of a clear definition of the users' needs and a comprehensive identification of the problems to be addressed. The characteristics of the problems, the specification of the knowledge required, and the expected benefits should be accurately analyzed and discussed for the successful set up of the next conceptualization stage.
Such aspects are herein debated, holding the discussion form two different points of view, i.e. from both the clinical problem and the system perspectives, leading to the definition of the functional and non-functional requirements of the HEARTFAID CDSS (ISO-9126, 1991).
In both cases, a user-needs focus is maintained and the final aim of responding to those needs as they develop, change, and mature over time is pursuit.
In deed, the problem domain of the HEARTFAID CDSS has been identified as the routine clinical practice in HF patients' management, and specialized considering the "*Problem Statements*", reported in the Deliverable D5 (section 8: *Problems Statement and formulation of the relevant Decision Making Problems*), as the set of decision making problems the system should solve.
Furthermore, general characteristics, related to system efficiency, portability and adaptability, are highlighted and discussed as constraints on the design and implementation.

6.1 HEARTFAID Specificity: Clinical Problem Requirements

As extensively reported in the introduction, HEARTFAID CDSS is mainly devised to intelligently support clinical operators in the daily management of HF patients. Actually, HF routine practice activities, e.g. diagnosis, prognostic evaluation and therapeutic planning, present several aspects in which an automatic, computer-based support could have a favourable impact (Perlini *et al.*, 1990; Tierney *et al.*, 1995; Wang *et al.*, 2000; Tierney *et al.*, 2003).
The accurate studies performed within the HEARTFAID WP1, "*Heart Failure Domain Analysis*", resulted in rigorous identification, formulation and assessment of all the clinical processes followed by decision makers in the HF domain, which would take advantages from computer-based interventions.
In order to better identify the role of the HEARTFAID CDSS in this domain, it appears useful underlining in which stages the system support is required, according to the HF care management program as introduced in the deliverable

[2] Please note, from this chapter on forth, we will use the word "system" and acronyms "CDSS" and "HEARTFAID CDSS" as equivalent, i.e. referring them to the HEARTFAID decision support system. When confusion could arise, an explicit meaning will be reported.

D8 - "*Definition and formulation of the organization and management models for the healthcare delivery*".

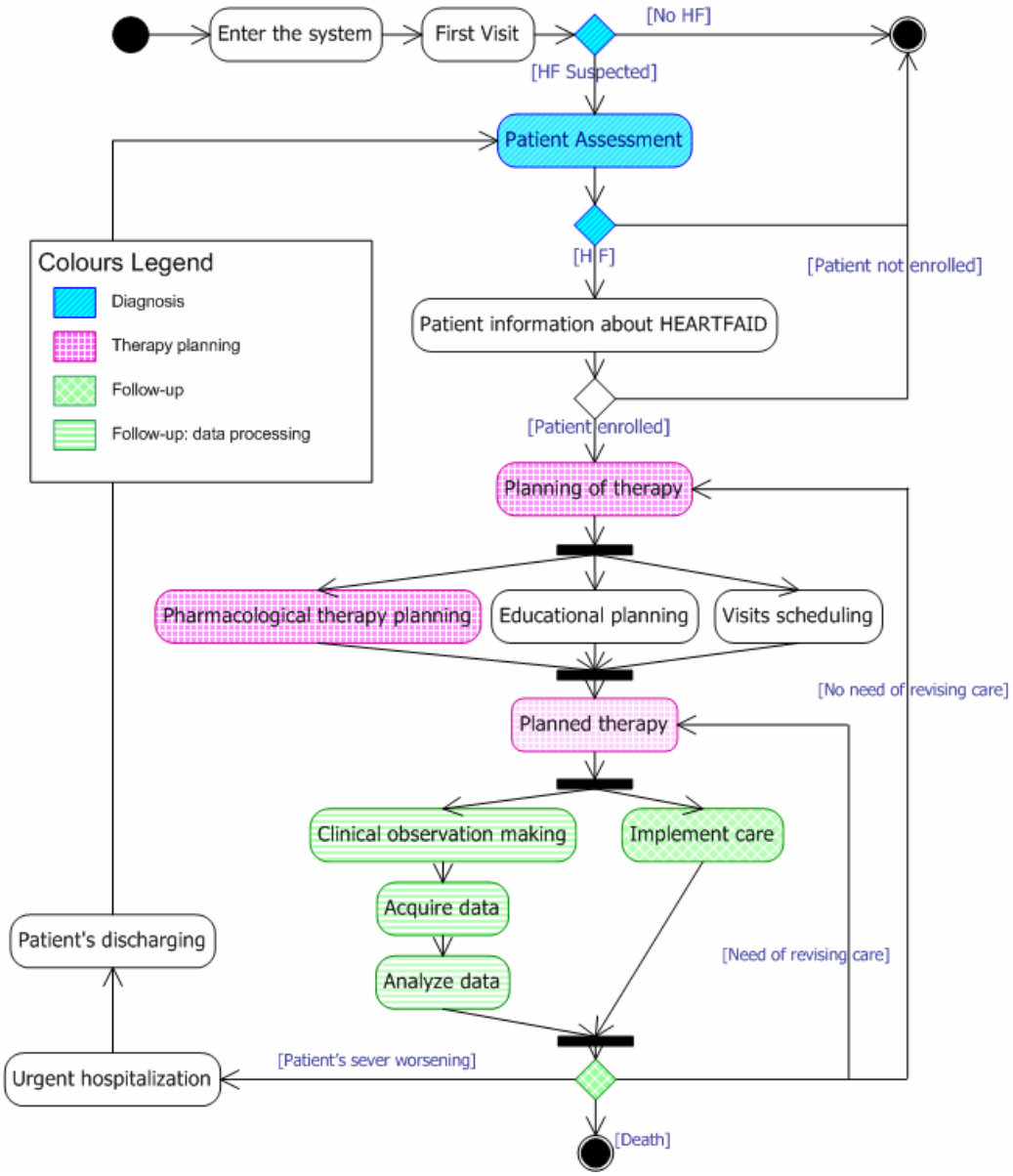


Figure 6-1. *The HEARTFAID Care Management Program Diagram: the different macro domain problems are highlighted colouring their steps. Diagnosis and prognosis problems are here included inside the same step, i.e. patient's assessment. Within the follow-up, a diverse texture is assigned to steps requiring data processing support (see legend inside the figure).*

In Figure 6-1, the UML diagram of the HEARTFAID *Care Management Program* activity (HFCMP) is shown. It sketches the macro-processes involved, i.e.

- HF diagnosis and assessment of the seriousness of the disease;
- virtual care team creation with patient's enrolment in the program;
- creation of the plan of care;
- training and education of the patient enrolled in the program;
- follow-up;

and the different phases constituting them.
The phases that would benefit from the HEARTFAID CDSS point-of-care interventions have been highlighted with different colours and textures according to the diverse macro-processes they belong to (see the legend inside the figure). From the decision support point of view, the macro processes can be referred as *macro domain problems* and listed up as (i) HF diagnosis, (ii) prognosis, (iii) therapy planning and (iv) follow-up.
More detailed decision problems can be identified for specifying the macro ones, i.e.

- HF diagnosis, assessment and severity evaluation
- Identification of suitable pathways
- Planning of adequate, patient's specific therapy
- Analysis of diagnostic exams
- Suggestion of changes in management and treatment
- Early detection of patient's decompensation

In order to focus as much as possible on the medical users' needs and the questions they indicate as compelling, even more specific decision making problems have been pointed out, according to what stated in the Deliverable D5, "*Medical-clinical processes and requirements in HF domain and formulation of the decision making problems*", (in particular in Section 8 of the same document – *Problems Statement and formulation of the relevant Decision Making Problems*). Some of them strictly concern the routine practice of HF management, e.g.

> «**Problem statement 1.** HFP, by collecting, integrating, and processing biomedical data relevant to HF diagnosis may overall increase the number of correct (positive or negative) diagnosis by decreasing both the number of falsely negative and falsely positive diagnosis […]»

whereas others pertain to a research environment devoted to the extraction and validation of new information, extending the current knowledge and expertise about HF management, e.g.

> «**Problem statement 2.** HFP may help in finding the single biomedical parameter or the combination of parameters (among signs, symptoms, and test results) that best identify subjects affected by HF»

The latter type of problems constitutes the target of the HEARTFAID KDD processes, which are completely devoted to the extraction of valid, novel,

potentially useful and understandable knowledge and to the definition of innovative decision models, on the basis of the information acquired during the project lifetime. The *discovered* knowledge and decision models will be integrated into the CDSS knowledge bases and used to solve the proposed problems. KDD processes are also provided for addressing new arising problems, making the HEARTFAID CDSS adapt to users' needs changes.

The decision problems, related to the daily clinical activity, has been selected as the starting point for the HEARTFAID CDSS development[3] and a "conversion work" has been performed in order to identify the corresponding requirements of the same system.

In the next section, such requirements are individually described, pointing out the problem characteristics, the required knowledge, and the expected benefits

By the way, before going in detail, some general considerations are noteworthy.

Mostly, the knowledge required by these problems is carefully reported in the available clinical guidelines for chronic HF management and treatment (Guidelines of the European Society of Cardiology – ESC, and of the American College of Cardiology/American Heart Association – ACC/AHA), which have been summarised in the Deliverable D5. Such knowledge can be formalized using one of the representation formalisms introduced in Chapter 3, and stored in the HEARTFAID CDSS knowledge base.
However, computational approaches appear also beneficial in most of the problems to be addressed: mathematical models and *Machine Learning* (ML) methods allow facing those decisional problems whose solution cannot be formalized in *symbolic* representations, owing either to the lack of assessed knowledge or to the characteristics of the problem, i.e. high levels of complexity and variability. Different studies demonstrated both the necessity and the benefits of such approaches within HF care management (Ortiz *et al.*, 1995; De Marco et al, 1997; Atienza et al., 2000; Wong *et al.*, 2003). In some cases, actually, making a decision requires an investigation on the hidden, complex, often non-linear correlations among data, together with high-level analytical processing functions. In such cases, the knowledge needed for the solution should be acquired directly from data (*inductive knowledge*) and stored in the developed model. ML *classification/prediction* and *regression* methods (e.g. *Artificial Neural Networks* – ANN, *Support Vectors Machines* – SVM) are examples of such computational decision models, which induce sub-symbolic knowledge from a data-driven processing.

[3] In fact, it is commonly agreed that once the problem domain has been defined for a DSS, the next task is to narrow the scope of the development effort by clearly defining the set of problems that the system will be expected to solve. The narrower the scope, the better are the chances that the DSS can be successfully built.

An overall benefit assured by the HEARTFAID CDSS can be easily identified as a consequence of the formalization and documentation of the HF domain knowledge, and the combination and formalization of the many experts' expertise. This, actually, will result in expanding the knowledge, improving decision making in the domain, and providing a wider distribution of the same knowledge in the clinical field.

6.1.1 Functional Requirements

For every macro process introduced in the previous section, the specific problems to be faced are herein discussed. The corresponding *Problem Statement* indicated in the deliverable D5 is explicitly referenced for each decision making problem. Besides, references to KDD processes are also reported when they are useful for the final decision making.

Diagnosis

Diagnosis activity in HF domain must be conducted via a holistic approach, considering both symptoms (fatigue, dyspnoea, etc.) and signs (in particular cardiac diseases evidences).
Diagnostic procedures are very well codified in HF Guidelines, but diagnostic errors are still present in the clinical practice. HEARTFAID CDSS should overall help physicians improving their diagnostic performance (*Problem Statement 1*). Thus, HEARTFAID CDSS should implement a method that accepts as input the patient status, and elaborate an answer about the accordance of the patient symptoms and signs with the diagnostic criteria.
The knowledge required consists in the evaluation criteria, specified in the guidelines, and in tacit rules that should be elicitated from the clinicians.
Confidence values should be evaluated and associated to the different passages of the diagnosis. In this way, more possibilities can be presented to the clinicians, supporting, in case, differential diagnosis.
Benefits consist of a more reliable diagnosis, by increasing the number of correct positive and negative diagnosis, while decreasing the number of falsely diagnosed cases.

In particular, HEARTFAID CDSS should also support the diagnostic capability of low resource medical settings, where not all of the expensive diagnostic equipments (e.g. echocardiograph) are readily available, and where the diagnosis of HF originates from simpler and less expensive methods (e.g. history and physical examination, ECG, chest X-ray) (*Problem Statement 4*).
In these cases, the CDSS diagnostic functionality should be able to elaborate an answer also in presence of missing information, underlining the possibilities stemming from the available data and suggesting some other examinations, if necessary for reaching a more reliable diagnosis. Computational decision models can be useful in these cases.

Besides, the CDSS may be of great support in classifying HF severity by guiding the physician through the established rules of NYHA classification (*Problem Statement 5*). In deed, according to the current guidelines, the management of HF depends on NYHA Classification (i.e. class of symptoms' severity), but NYHA symptom classification is subjective and difficult to separate from concomitant disease conditions. The HEARTFAID CDSS should, then, follow the clinician's evaluation process, by suggesting the different conditions to evaluate.
In this case, the knowledge required consists of the rules specified in the guidelines: they can be grouped in a *protocol*, i.e. a sequence of steps the clinicians have to follow, evaluating a number of subsequent conditions. Suitable knowledge representation formalism and interactive functionalities should be, then, selected and implemented.
If, form the KDD processes, new rules or decision models for HF severity classifications will result, the HEARTFAID CDSS should supply the results of both classification schema, i.e. the NYHA and the newly encoded.
The resulting benefits consist in an improved evaluation of the patient's status, which, in its turn, fosters the other patient's management stages. Table 6.I summarises the described characteristics.

Table 6.I. *Requirements for the diagnosis problems, in terms of HEARTFAID CDSS intervention, knowledge required and benefit assured*

Diagnostic Decision Problem	HEARTFAID CDSS intervention requirements	Knowledge Required	Benefit
Diagnosis	· **Diagnostic suggestions** · **Differential diagnosis** · **Confidence values**	· **Symbolic representation of guidelines and clinicians' expertise rules;** · **Computational decision models for computing discriminative parameters**	· **Reduction of false negative diagnosis;** · **Improvement of true positive diagnosis.**
Low-resource environment diagnosis	· **Diagnosis formulation** · **Suggestions on examination to perform** · **Missing values management**	· **Symbolic representation of guidelines and clinicians' expertise rules;** · **Computational decision models able of handling missing values**	**Knowledge sharing in different health care environment**
Severity assessment	· **Compliance of guidelines classification protocol** · **Suggestion of new classification**	· **Guidelines protocol suitably codified;** · **Symbolic knowledge or computational decision model for new classification**	**Improved evaluation of the patient's status**

Prognosis

Among the decisional tasks that physicians have to face during HF patients' management, prognosis evaluation is one of the hardest, given the absence of operative information in literature (Cohn, 1989; De Marco *et al.*, 1997; Atienza et al., 2000; Wong *et al.*, 2003). In addition, for each of the possible conditions (i.e. *stable*, *improving*, *rapidly worsening*, or *slowly worsening*) both parameters predictive value and prognostic evaluation objectives may change. Prognostic stratification during an acute unstable phase, for instance, should guide immediate decisions, while, in a stable phase, it could have a long term aim and should predict destabilization and death in mid and long term.
In this context, HEARTFAID CDSS should be able to formulate patient's prognosis, taking into account the implications of different patients' status (*Problem Statement 17*). Additionally, it should also help in verifying if the clinical classification of acutely decompensated HF adds prognostic value to the NYHA classification (*Problem Statement 12*).
Computational decision models for predictive analysis can be required to accomplish this task. Innovative techniques for survival analysis using ML and statistical methodologies should be then provided. Moreover, in the circumstances where new cut-offs, and/or reference values for individual, or combinations of parameters will be discovered by KDD processes, they should be suitably encoded into the HEARTFAID CDSS knowledge bases, thus allowing to a further increase of the efficacy.
The necessary knowledge consists partly of the rules that can be extracted from guidelines and clinicians' know-how, and mostly of inductive knowledge codified in the predictive decision models.
As consequent benefits, more reliable prognostic evaluation can be obtained, with an overall improvement of patients' management and outcome.
Table 6.II summarised all the characteristics of this problem.

Table 6.II. *Requirements for the prognosis problems, in terms of HEARTFAID CDSS intervention, knowledge required and benefit assured*

Prognostic Decision Problem	HEARTFAID CDSS intervention requirements	Knowledge Required	Benefit
Prognostic evaluation	• **Prognosis formulation;** • **Different prognostic values in different patients' conditions**	• **Some knowledge from clinicians' expertise;** • **Computational decision models based on ML and statistical methods;** • **Predictive parameters extracted from KDD processes**	• **Improvement prognosis reliability;** • **Fostering patients' management and outcome**

Genetic analysis

HEARTFAID CDSS should provide a service able to evaluate patient's genetic data, in order to provide risk evaluations. In particular, even if HF Guidelines don't contain specific information on the argument, starting from the results available on the literature and exploiting HEARTFAID data mining studies, it should be possible to implement a functionality for the analysis of the risk related to patient genes polymorphisms.

Therapy planning

HEARTFAID CDSS should suggest, based on guidelines recommendations, the pharmacological and non pharmacological measures that best apply to each individual patient (*Problem Statement 6-7*). Recent HF guidelines codify in detail therapeutics actions that should be performed with regards to different patients' status. However, the number of possible actions and the complexity of the whole context suggest the usefulness of an automatic tool able to remind the available therapeutics options.
In particular, the HEARTFAID CDSS should be focused on supporting the plan of pharmacological treatment and life style modification. Actually, daily clinical management of chronic HF patients is mainly related to drugs prescription and patients' status monitoring, while surgery actions touch cardiac disciplines enough far from HF treatment.
Moreover, pharmacological treatment is characterized by continue changes, in relation to patients' answer to the treatment: for instance, every new drug has to be up titrated until its optimal dose. Hence, pharmacological suggestions cannot be provided only on the basis of the patients' actual status, but also in relation to their complete clinical story, with particular regard to possible previous intolerances and allergies.
Taking into account all the above considerations, the HEARTFAID CDSS should provide assistance in planning the therapy by carefully considering HF aetiology, gravity, physiopathological conditions, and by evaluating interactions with other coexistent diseases and/or drugs taken by the patient.
The knowledge required consists mainly in the criteria inferable form the guidelines, and the benefits consists in an overall improvement of patients' management (Table 6.III).

Table 6.III *Requirements for the therapeutic planning problems, in terms of HEARTFAID CDSS intervention, knowledge required and benefit assured*

Treatment Decision Problem	HEARTFAID CDSS intervention requirements	Knowledge Required	Benefit
Therapy planning	• **Pharmacological treatment suggestions** • **Life style modification alerts;** • **Uptritation support;** • **Compliance to guidelines recommendations**	• **Prescription criteria inferable from the guidelines** • **Knowledge related to drugs interaction** • **Knowledge related to drugs tollerance**	**Improvement of patients' management and outcome**

Follow-up

Telemonitoring applications allow the repeated home collection of relevant symptoms and biomedical parameters like blood pressure, heart rate, respiratory frequency, and so forth.
Within HFP, the final aim of telemonitoring is avoiding frequent patients' hospitalizations via the prevention of adverse events and the early detection of sudden worsening. This can be achieved by suggesting changes to the therapy, advising new diagnostic examinations and early detecting patients' decompensation.
Even if telemonitoring technology is quite mature, and allows the collections of an impressive number of different biomedical signals, the automatic recognition of decompensation events is an absolutely innovative functionality. To this end, the HEARTFAID CDSS should infer on the available symbolic knowledge should be formalized to let the CDSS reason on it. Perhaps more reliably, computational decision models should be implemented for processing a set of telemonitored symptoms and signs, and assessing the current state of the patient. Suitable suggestions should then be provided according to the achieved evaluation.
As introduced, the benefits will mainly consist in an optimization of the therapy, a better outcome and a reduction of the health costs (Table 6.IV).

Table 6.IV. *Requirements for the follow-up problems, in terms of HEARTFAID CDSS intervention, knowledge required and benefit assured*

Follow-up Decision Problem	HEARTFAID CDSS intervention requirements	Knowledge Required	Benefit
Adverse events prevention	• Biomedical signs processing; • Therapy changes suggestions; • New diagnostic examinations suggestions	• Symbolic representation of guidelines and clinicians' expertise rules; • Computational decision models for evaluate acquired biomedical signs	• Optimization of the therapy; • Improvement of patients' outcome; • Reduction of the health costs
Early detection of patient's decompensation	• Biomedical signs processing; • Alarms	Predictive decision models able to for evaluate acquired biomedical signs	

6.2 HEARTFAID CDSS System Requirements

Besides the functional requirements to satisfy, the HEARTFAID CDSS should also enjoy important properties that are required for assuring its usability (i.e. approval by the users, and successfully integration inside their routine activity), and necessary for guaranteeing its implementation effectiveness. Moreover, some advisable characteristics can be pointed out also for fostering the success of the overall HEARTFAID project and its capability of growth.
In the following, such properties, usually defined as *non-functional* requirements, are introduced and discussed according to the (i) users' interaction, (ii) system design and (iii) project extension point of views.

6.2.1 Users Interaction Requirements

A good literature exists about key success factors of clinical decision support systems (Holbrook *et al.* 2003; Coiera 2003; Kawamoto *et al.*, 2005; Zheng *et al.* 2005), as reported in chapter 3. Beyond other characteristics that can be implemented only at end user level (i.e. user-friendly interface), there are some issues that should be taken into account since the planning of the support system.
In particular, the following properties have to be underlined as desirable.

Suggestion completeness
A CDSS suggestion should not be provided "as it is", that is as the only possible alternative without any other choices. Instead, every suggestion should report a list of possible alternatives, in order to allow physicians to make their decision. Every component of "choices list" should be accompanied by a reliability value, in order to suggest a useful ranking.
Suggestions completeness is particularly essential for both prognostics and therapeutic decisions. During prognostic evaluation, all the possible outcomes

should be presented to the clinicians, each one with the respective likelihood. In this way, physicians have a more precise and comfortable vision of the various possible evolutions of patient status, and can easily approve or critic CDSS results.
Besides, therapeutic planning has to face several constraints that can afflict the validity of CDSS suggestions. In circumstances when a patient has to assume a particular drug that is not compatible with the best treatment suggested by the guidelines, another therapy should be prescribed. A great point of strength of HEARTFAID CDSS would then be represented by the automatic provision of the entire set of possible treatments, indicating the best one but including also the non optimal ones.

Interactivity
Interactivity is another requirement with high adding value for HEARTFAID CDSS. Often, physicians have to get two or more decisions consequently, and each decision relies on the previous ones. Interactivity is, thus, needed in order to implement this kind of mechanism, by a dialogue between the user and the CDSS (e.g. structured as a "question and answer" sequence).
Interactivity feature could be implemented for almost all the functionalities of the CDSS, but it is particularly important for diagnostic evaluation and therapeutic planning.
The responding smartness and rapidity represent other advisable properties of the system, which imply that design and implementation efforts should be devoted to reduce the computational complexity of the same.

Suggestion explications
The availability of exhaustive explication function is mandatory for assuring the success of the system. In practice, every time a suggestion is formulated, it has to be accompanied by a comment explicating why that particular suggestion has been produced. For instance, when suggesting the prescription of ACE inhibitors, a comment, quoting the correspondent guidelines recommendation and the patient's condition that entail the same prescription, could be much appreciated.
When suggestions are generated by inferring the knowledge base, explanations are easily derived from the same (e.g. listing up the comments inserted in the knowledge representation formalism). Whereas, if the suggestion is obtained by applying one or more computational decision models, e.g. an ANN, explanation could illustrate the origin of suggestion and present the reliability score of the prediction obtained.

6.2.2 System Design Requirements

Requirements belonging to this category concern constraints on the system design and/or implementation, regarding important aspects such as its reliability, quality, security, and so forth.

Modularity

Developers and managers are usually interested in handling simple, robust, easy to control, and easy to modify systems. Modularity has a fundamental role in pursuing such properties.

A modular design consists in the identification of the different components of the system that are responsible of the various tasks to accomplish, and can be easy assembled, used and repaired. Such components should separate cleanly and fit together well, so that modifying one of them will have no adverse affects on the others. High modularity costs some design time, but pays back well through clarity, elegance, maintainability and flexibility.

For satisfying this property, the CDSS should be composed by independent, reliable components with the primary goal of separating knowledge and data from processing units. Moreover, different types of knowledge, data and applications should be identified and suitably organizing so to avoid ambiguity, and assure the system clarity and flexibility.

Standardization

In the clinical field, the importance of standardization is evident and over and over pursuit. A standard is therein intended as a set of mandatory requirements employed and enforced to prescribe a disciplined, uniform approach to processes and operation development. Conformity to a standard assures several advantages that can be summarised as achieving and maintaining the most effective levels of compatibility, interoperability, interchangeability and commonality.

The standardization of the HEARTFAID CDSS means adopting, or at least referring, validated, well-established standards for knowledge representations, reasoning processes development, data processing and communication. This will assure a wider diffusion and easier deployment of the system, and the possibility of sharing and reusing knowledge and decision support applications.

Moreover, a standard CDSS fosters the standardization of the entire HFP. This is a fundamental demand, since the HPF will act in the Babel of healthcare structures, where data accessing, exchanging, integration and interoperability are overwhelming necessities.

Portability

Since the HEARTDFAID CDSS will be deployed in several validation sites, each characterized by the availability of different devices and settings, assuring its portability is compelling.

Portability should be meant as a measure of system independence from the environment, with the consequence possibility to transfer it without making relevant adjustments or limiting its functionality. Having a portable HEARTFAID CDSS requires adopting portable software implementation languages, and assuring its functionalities with various operating systems, databases, and networks.

Open source
An advisable requirement is implementing the HEARTFAID CDSS by using open source software tools.
In deed, this would assure a number of advantages. First of all, the software license of these tools is mostly free, or anyway less expensive than the relative commercial one. The software quality is in general assured, given the high number of qualified experts collaborating to its development, tests and documentation (for sure higher than a single commercial programming group). Besides, thanks again to the impressive number of developers, version updates are quick, and the code stable and secure.
Usage continuity will be assured by the developers' community while this cannot be the same for commercial software, which depends on the state and choices of the producer (i.e. based on continuing profitable and more diffused products).
As greatly desirable for HEARTFAID, open source software is often portable, that is it is allowable for different operative systems and in different languages and for different hardware. Finally, it is also more reliable than proprietary one: checked and verified by developers' communities.
All the above mentioned advantages are of great value in developing the HEARTFAID CDSS, since the system will be used in different sites, with different devices and settings, and liable to changes.

6.2.3 Project Requirements

HEARTFAID stands as a highly innovative platform of services able to assure (i) the optimisation of medical processes, (ii) an important increase of the treatment quality as well as (iii) a remarkable reduction of healthcare costs. Such characteristics make the HFP an example to follow, a *template*, for the management of several clinical problems.
It might be, then very easy and profitable spreading the use of its family of services to

- the largest number of clinicians and health authorities, for improving the specific treatment of the heart failure;
- other healthcare communities.

In order to reach these goals, the HFP and thus its CDSS should enjoy the following properties:

- **scalability**, that is the ability to handle increased volume or complexity, in terms of number of users, amount of data, processing workload;
- **flexibility**, corresponding to the possibility of incorporating information from diverse sources;
- **adaptability**, that is the ability to adapt to varied practice settings and to reflect changes occurring in the medical domain;
- **extensibility**, which permits easy additions of new knowledge and services.

The following table (Table 6.V) summarised all the properties the HEARTFAID CDSS should enjoy.

Table 6.V *HEARTFAID CDSS Non-functional Requirements*

Users' Interaction Requirements	System Design Requirements	Project Extension Requirements
❖ **Complete on important issues** ❖ **Easy to communicate with** ❖ **Quick to response** ❖ **Understandable**	❖ **Modular** ❖ **Standard** ❖ **Portable** ❖ **Open source**	❖ **Scalable** ❖ **Flexible** ❖ **Adaptable** ❖ **Extensible**

7. Functional Specifications of the HEARTFAID Decision Support System

In the previous chapters, we have investigated the methodological foundations of decision support with particular attention to clinical applications, analyzed the specificity of the HEARTFAID CDSS and stated the requirements necessary to satisfy the HEARTFAID users' needs (*Identification* stage). In this chapter, we aim at referring all the previous discussions to the functionalities the HEARTFAID CDSS should provide to fulfil such requirements (*Conceptualization* stage).
Starting with a description of the HEARTFAID domain, for better focusing the discussion on the role of decision support, we firstly specify the main characteristics of the HEARTFAID CDSS with respect to its *functional position* inside the HFP. An insight into the system is, then, given by describing its functionalities and the architecture designed for supplying them.
Some use cases are also discusses for showing the system in action.

7.1 HEARTFAID Domain

The HEARTFAID CDSS represents the core of the HFP of services, as the main responsible of making more effective and efficient all the processes related to HF patients' diagnosis, prognosis, therapy and follow-up.
All the functionalities and services supplied by the entire HFP, as a complex clinical information management system, have been sketched in ***Figure 7-1***. As highlighted, such functionalities can be further grouped into three macro *contexts*[4]: *data collection and management*, *knowledge-based decision support* and *end-user applications*.
As a whole, the HFP has a transversal value with respect to the DSS categories described in chapter 3. Thus, introducing the above mentioned contexts appears useful for better describing the platform functionalities, and identifying their responsible components. In this perspective, the functionalities pertaining to the end-user applications context fall in the category of the *Group* DSS, allowing the interaction among the several HF stakeholders as identified in the Deliverable D8, i.e. care givers, specialized nurses, general practitioners, specialised doctors, patients. Furthermore, the characteristics of the *Web-based* DSS category might also be ascribed to this context, since a web-based portal will be provided as the doorway to a multitude of end-user utilities and services.
The data collection and management context emphasizes access to and manipulation of structured patients' data, supplying the main functionalities that belong to the *Data-driven* DSS category. Key importance should be recognized, in this context, also to the KDD services, which exploit a data warehouse for

[4] A context is herein considered as a set of functionalities or services devoted to a unique, high-level scope.

applying advanced models and methods tailored to the extraction of new usable knowledge.

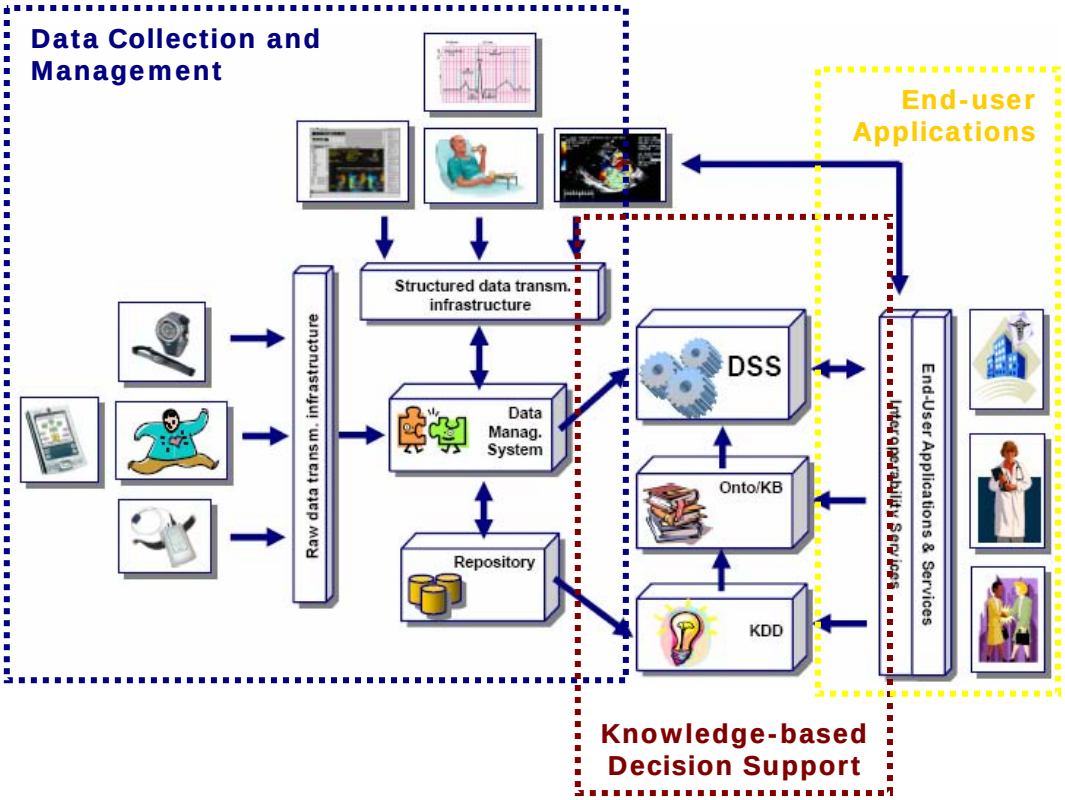


Figure 7-1. *HEARTFAID System Functionalities and Services*

As a *data mining* approach, such service plays a strategic role within the decision support context as well, as source of *inductive knowledge*. The latter context of services stands as the *core* of the HEARTFAID platform. Its peculiarities obviously coincide to those of the *Knowledge-driven* DSS category: the knowledge base constitutes the dominant component, formalizing physicians' expertise and know-how, and the best practice approaches (clinical guidelines) about HF diagnosis and management (*deductive knowledge*). In particular, since computational decision methods can result from KDD processes for solving hard HF clinical problems, a *Model-driven* connotation can be also ascribed to this level, considering a decision model as an abstract representation of a decision problem in terms of variables and processing functions for obtaining its solution. Moreover, as detailed in the previous chapters, data processing algorithms are devised for extracting useful information from diagnostic signals and images. They can be integrated into the decision support services as analytical models as well.

This twofold value represents an innovative feature of HFP decision support service: it supports clinical decision making by hybridizing (*symbolic*) knowledge based inference with advanced computational (*sub-symbolic*) decision models.

In such a complex functional scenario, a virtualized, logic view of the platform turns out useful for simplifying and optimizing the management, orchestration and provision of functionalities. This view is mirrored in the overall HEARTFAID multi-level, heterogeneous and distributed architecture, shown in Figure 7-2 (introduced in the first chapter and herein reported for the sake of clarity).

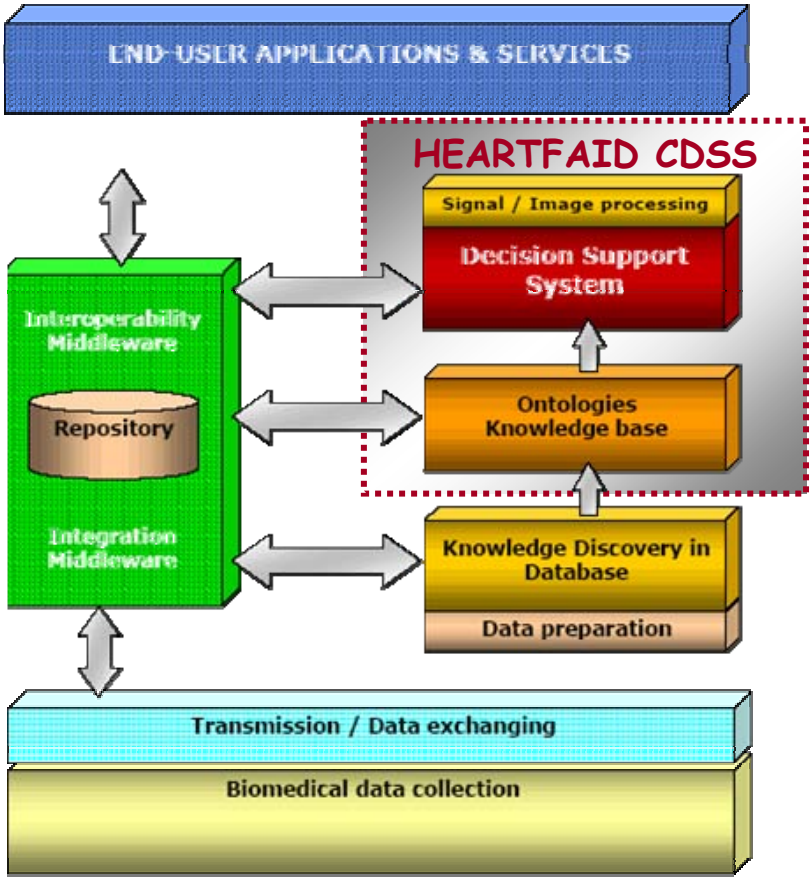


Figure 7-2. *The HEARTFAID Architecture*

The different components of the platform can be seen as *resources*, *virtualizing* the operations required for their management. At the application moment, the involved components are dynamically integrated for supplying sophisticated but much flexible applications.
The HEARTFAID CDSS component is a platform resource as well, able to offer a number of functionalities and to interact with the other resources for performing its tasks.
The responsible of guaranteeing integration and interoperability among all the HFP components, as well as the services provided to the end-users is the HEARTFAID Middleware. Its functional details have been extensively described in the deliverable D11, "*Functional specifications of the Middleware*", illustrating both data and processes flows among the platform components, and the approach to resources management (Figure 7-3).

The action sphere and the functional requirements of HEARTFAID CDSS resource have been identified in the previous chapter. Summing up, four macro domain problems have been pointed out (i.e. heart failure diagnosis, prognosis, therapy planning and follow-up) and, for each of them, precise decision making problems have been asserted, according to the *Problem Statements* detailed in the deliverable D5. Such precise identification allows to carefully tuning the supplied functionalities on the users' needs.
Each problem can be conceived as a *request* or a *class of requests* committed to the HEARTFAID CDSS, which is then activated *on-demand*.

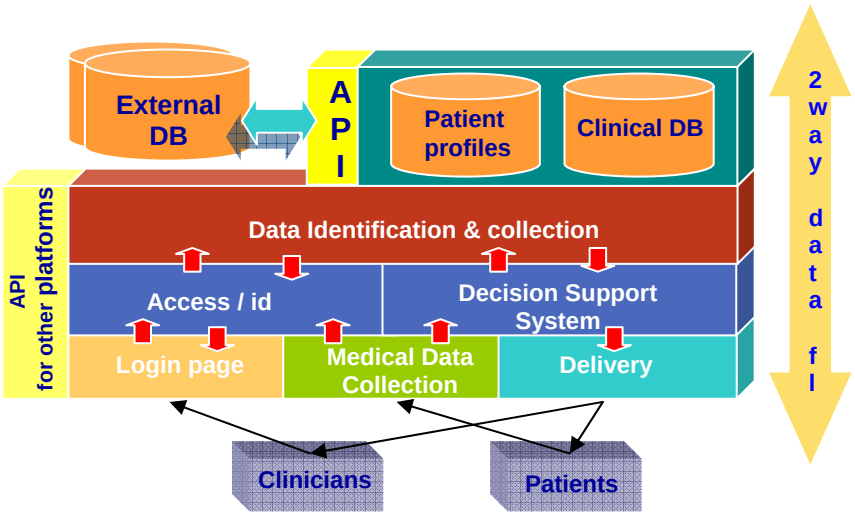


Figure 7-3. *Integration of several HFP functionalities: interaction of the end users with the main components/functions of the central middleware.*

The CDSS handles every request according to a specific encoded policy, interacting, when necessary, with components belonging to the other HFP contexts (e.g., the end-user application context for interacting with the user, or the data management context for retrieving and storing patients' data). This means that the system should represent an *active* component of the platform, able to demand requests in its turn as well.

An insight into the HEARTFAID CDSS functionalities for requests management is provided in the next sections.

7.2 HEARTFAID CDSS Functionalities

The capabilities of the HEARTFAID CDSS can be grouped into the following main categories:

Reasoning on patients' data

The CDSS solves the requests of supporting HF management, i.e. diagnosis, prognosis, therapy and follow-up, by using patients' heterogeneous information (i.e. actual status, anamnesis, clinical history, diagnostic parameters, clinicians' evaluation) and applying two type of reasoning: *inferential* and *computational*.
Inferential reasoning consists in *inductive* and *deductive* reasoning on the available domain knowledge, and it is strictly correlated to the knowledge representation formalism. Several approaches appear appropriate for formalizing HF domain knowledge. The fundamental aim is developing a formal description of the domain (i.e. relevant concepts, their properties and interrelations) and carefully formalizing the declarative and procedural knowledge, derived from the guidelines and the experts' know-how. Ontologies combined with rules seem the more suitable and up-to-date methodology for solving both tasks. Actually, rules-based approach appears the more appropriate, since easily understandable by a non-specialised audience, e.g. clinicians. In this way, they can be more tempted in contributing to the knowledge elicitation and representation processes. An inference engine can then be devised for the corresponding reasoning processes. However, it can happen that, according to guidelines recommendations, satisfying a request entails the compliance to specific clinical *protocols*, i.e. plans of successive *condition/evaluation/action* steps, e.g. assessing patient's HF severity according to the NYHA classification. In these cases, a reliable representation may be adopted, e.g. by simply connecting the corresponding rules, or by explicitly modelling protocols as *activity plans*, or by using *workflows* when the protocol is highly complex. The role of the CDSS consists, in these cases, in guiding the clinician, throughout the protocol steps, and assuring their compliance.

Computational reasoning is involved in those difficult HF decision problems whose solution is still debated in the medical community, due to the lack of validated and assessed evidences. It consists of *inductive reasoning* and relies on applying computational decision models, which can be developed according to different methodologies (e.g. ML including *Soft Computing* and *Statistical Pattern Recognition*, or *Linear* or *Constraint Programming*).
Defining ML decision models requires the identification of a set of input parameters for modelling the problem to be faced; a set of *exemplars* of the problem, and a *training* process for data-driven knowledge extraction. This process relies, in practice, in tuning the *model free parameters*, iteratively until the desired decision accuracy is reached.

CDSS supports decision making according to two *modalities*: proving hypothesis, e.g. diagnosis formulation hypothesis, or inducing a decision starting from the data. Such functionalities make possible both differential and evidence-based diagnosis processes.
As underlined in the requirements, uncertainty and missing values are, also, managed, associating confidence values to the final suggestion.

Summarising, from the users' point of view, such functionalities can be implemented for allowing the clinicians (i) formulate their diagnosis and exploit the CDSS to prove them; (ii) query the system for obtaining diagnosis or prognostic evaluation; (iii) exploit the system for the prescription of the most appropriate therapy (in deed, the system is able to process all the clinical information about patients, available pharmacological treatments and their contraindication and inadvisability).

Pattern recognition in historical patients' data

The CDSS provides the clinicians with pattern recognition functionalities in the historical data of the central repository. In this way, they can recover specific information and clinical cases similar to a predefined pattern.
The final goal is supporting the *case-based reasoning* of clinicians, who are looking to already treated cases for making decision regarding the actual one. In deed, using this service, the clinicians will be able to visualise all the clinical information of the patients that have a certain level of similarity with a specific case study under examination.
Two modalities are provided, i.e. automatic and not automated retrieval. In the latter case, the pattern to retrieve is selected by the user, thus a simple search in the repository, using a similarity measure, is required[5]. In the former, the system autonomously retrieves the similar cases by applying pattern recognition (e.g. *clustering*). In particular, such functions can be useful also for managing exceptions, i.e. when the system cannot suggest a decision according to its knowledge base or applying its models (due to too many reasoning conflicts or poor confidence values). In such cases, retrieving and showing a case similar to the one at hand can be a solution.

Signal and image processing

The specific data processing methods, devoted to extracting relevant parameters from acquired signals and images, are supplied as important facilities of the CDSS. They are useful for supporting diagnostic activities in routine clinical practice, and fostering the discovery of novel, significant knowledge in the HF domain. Since applied for computing important diagnostic parameters, they provide valuable contribution to both clinicians' daily activity and CDSS reasoning processes.
The functional specifications of such algorithms have been extensively debated in Chapter 5. Here, it is important to underline that they can be applied for answering to a specific incoming request, i.e. from the users that would like to assess some parameters, or while accomplishing a more general request of decision support.

[5] In such case, this functionality could be delegated to the repository managing system.

Patients' telemonitoring

Patients' telemonitoring is a complex task that involves all the HFP functionalities contexts. Within the CDSS purview, it requires the reasoning functionalities already introduced, but is herein separately discussed for its peculiarities.
The CDSS can be used for detecting specific conditions or patterns in the monitored data that might indicate patients' critical situations. After a first pre-processing, both inferential and computational reasoning are applied, taking into account data trends and concurrent conditions that might influence the patients' status. When an adverse event is detected or predicted, the system decides to send alarms or alerts according to the gravity of the situation detected and the consequence risks for the patient (an end-users application is responsible for the effective dispatch to clinicians and patients).

For better specifying the introduced functionalities, in the forth, the system functioning is explored from different points of view, i.e. from

a. the data perspective,
b. the functional perspective,
c. the control perspective,

Such perspectives can be summarised as the "*what, how,* and *when*" view of the system.
The same approach is followed in the next section for the description of the system architecture, by detailing the HEARTFAID CDSS building-blocks, i.e. its components, how they are assembled and what relations exist among them.

The heterogeneous patients' data managed by the CDSS fall into two main categories:

a.1. Information *already* evaluated: qualitative data related to patients' status, anamnesis, clinical history, and other quantitative parameters already extracted from diagnostic examinations, e.g. by the devices for signals or images acquisition, or as clinicians' considerations. They have to be considered input data, inserted from the clinicians, retrieved in the repository or obtained from the acquisition devices;

a.2. Information *to be* evaluated: qualitative evaluations and quantitative parameters that the system should perform or estimate, e.g. diagnostic evaluations; predictive parameters; representative features extracted from signals and images. They have to be considered outputs of the system, which will be stored in the repository and afterwards used, in their turn, as system inputs, when required.

Other relevant data, not strictly related to the patients, consist in general information about, for instance, drugs availability.

Processes involved in the CDSS functioning can be grouped in the following categories:

b.1. *Signal and images processing*, developed for assessing patients' status and acquiring diagnostic parameters. Processes belonging to this category consist of calls to algorithmic procedures, are activated on demand, and might require the user's interaction, e.g. procedure for estimating the ejection fraction in echocardiographic images;

b.2. *Inference*, involved in inferential reasoning, for assessing patients' status, formulating diagnosis and prognosis, assisting therapy planning, and patients' monitoring. Generally speaking, they consist in querying the knowledge base for inferring new facts or new actions to perform, are activated on demand and can involve the users' in interactive dialogues;

b.3. *Computational decision*, concerned with computational reasoning for patients' HF severity and prognosis assessment, and for patients' monitoring. They mainly correspond to on-demand applications of the decision models developed.

b.4. *Pattern searching*, aimed at retrieving clinical cases similar to current one, by applying pattern recognition methods.

The system control can be discussed according to two points of view:

c.1. *HFP perspective*: as introduced in the previous section, the HEARTFAID CDSS is activated on demand, after the commitment of a request. Once activated, it has the total control of its processing activities, and can interact with the external component of the HFP, committing requests in its turn, when necessary.
The proactivity of the system, i.e. its capability of preventing patients' adverse events, is provided during telemonitoring, by formalizing suitable reasoning processes. Anyway, CDSS services delivery remains on demand.
The prevention of other critical situations may be performed by formalizing the *pathways* that the patients have to follow, i.e. therapeutic prescriptions to respect (which drugs, how much and for how long); diagnostic examination to perform (when and where); and so forth. A specific HFP component, an *Agenda*, may be inserted in the end-user applications context and verify the pathways compliance, by sending reminders and alerts to both clinician and patient. This entails checking all the possible critical situations in advance, when planning patients' management, e.g. establishing drugs dosage and therapy duration in careful respect to the patient's peculiarities, and planning periodic revisions of patient's answer to treatment.

c.2. *CDSS internal perspective*: suitable processing strategies would be implemented for supplying systems functionalities. As we will see in the following sections, appropriate *strategies*, for selecting the type of reasoning process and managing its application, are encoded and followed,

e.g. when supporting the patients' prognostic evaluation, both inference on patients' data and predictive decision models can be available, the CDSS, then, applies both the processes and suitably combines their results for providing its final response.

7.3 HEARTFAID CDSS Modelling

The complexity and diversity of data, processes and tasks involved, require a careful conceptualization of the CDSS in order to assure its reliability.
Once described its functionalities, modelling the CDSS is the next, fundamental step of its conceptualization. It consists in identifying the tasks that the system should address, and its logical-functional structure, i.e. its modular organization in independent components, their relations and functionalities.
System tasks concern managing external requests, which can require different CDSS functionalities. According to what discussed till now, some evident examples can be outlined: they are formulated in descriptive, non-rigorous form, and reported in Table 7.I, together with the CDSS tasks, functionalities demanded, and a rough, high-level description of the *sub-tasks* or *actions* performed. Although rough, the description serves the scope: highlighting that CDSS application is not a simple inference of new facts, but entails a number of other correlated operations, starting from patient's data loading to the combination with computational decision models. A more detailed description would require expanding each sub-task, listing up the required actions.
For modelling the CDSS, three fundamental levels have been identified:

- the *knowledge level*, corresponding to all the information needed by the system for performing tasks, e.g. data, domain knowledge, computational decision models;
- the *processing level*, consisting of the system components that are responsible of tasks accomplishment by using the knowledge level;
- the *end-user application level*, including the system components whose functionalities are specifically defined for interacting with the user.

Besides, the knowledge level has been further specialized in three sub-levels, i.e.:

- *Strategy level*, concerning the knowledge about the tasks performed by the system, e.g. their decomposition in sub-tasks or actions, as well as their application order, and when and under what conditions they have to be performed.
- *Reasoning level*, composed by the declarative, procedural and computational knowledge used by the system for performing its task.
- *Domain level*, containing the formalization of the domain structure, which has to be known to make inferences and execute tasks.

Such conceptualization division makes the organisation of knowledge inside the system explicit, providing an implementation-independent description of the role that various knowledge elements play during the decision supporting process.

Table 7.I. *Simple examples of requests committed to CDSS, with a description of their management, i.e. tasks, functionalities and actions. The step "Elaborate suggestions on next actions" is optional for the first request, but can increase CDSS value.*

End-user Request	CDSS Task	CDSS Functionalities	CDSS Actions
R1 «Check the validity of HF diagnosis»	Prove HF hypothesis	Inferential reasoning on patient's data	▪ Collect patient's data relevant for diagnosis; ▪ Query the KB; ▪ Return results; ▪ *Elaborate suggestions on next actions*
R2 «Is the patient's status alarming?»	Assess patient's status (during monitoring)	Predictive model, supported by Inferential reasoning	▪ Compute model inputs; ▪ Apply the predictive model for estimating the patient's status class; ▪ Return results (if true, alerts/alarms); ▪ Elaborate suggestions on next actions, by querying the KB
R3 «How much severe is patient's heart failure? »	Compliance to guidelines *protocol*	Inferential reasoning	▪ Execute *plan* for following the protocol, by establishing a dialogue with the user
R4 «What's the patient's prognosis»	Prognostic evaluation	Predictive model, and Inferential reasoning	▪ Compute model inputs and collect patient's data; ▪ Apply the prognostic model for estimating the patient's status class; ▪ Infer on KB ▪ Combine both results and supply the final response ▪ Elaborate suggestions on next actions, by querying the KB

In particular, this conceptualization induces a corresponding division of also the processing level, which is, then, composed by

- a *meta level*, devoted to task accomplishment and sub-tasks organisation;
- an *object level*, responsible of performing sub-tasks, by reasoning on the reasoning and domain knowledge.

Figure 7-4 shows this *multilevel* structure of the knowledge and processing level of the system.

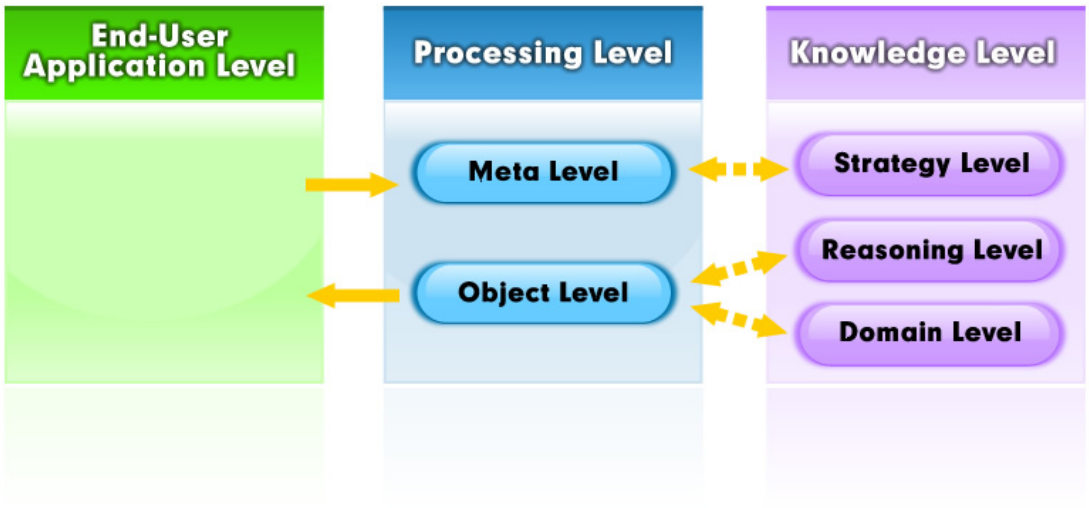


Figure 7-4. *The conceptualization levels defined for modelling the CDSS*

Next section details how this conceptualization has been mapped onto the CDSS architecture.

7.4 HEARTFAID CDSS Architecture

The HEARTFAID CDSS architecture has been designed according to the above multilevel conceptualization. Besides the modelling elegance and the advantages already discussed, different reasons have furthered this approach: first of all, the availability of different types of knowledge, also within the same level, i.e. the formalized HF knowledge base, and the computational decision models. They have, of course, completely different structures and allow the applications of different reasoning processes (see section 7.2), but have overlapping functionalities, i.e. they can be used for answering the same request. Also data processing and pattern searching algorithms demand an appropriate management, since they can be applied upon explicit request or invoked during the reasoning. Other motivations are related to data access and management: due to its dimension, the patients' data repository cannot be considered as a component of

the knowledge base[6]; hence, they should be collected outside the CDSS. Basing on such considerations, a model consisting of a unique module for the entire organization of request management could be highly unfeasible. Despite, a two level model assures high reliability, thanks to the processing load partitioning. A *meta* processing level has been then designed for organizing tasks accomplishment, and for managing the requests from and towards the other HFP components. An underlying *object level* has, then, been intended for the reasoning processes.

More in detail, the CDSS architecture consists of the following components (Figure 7-5):

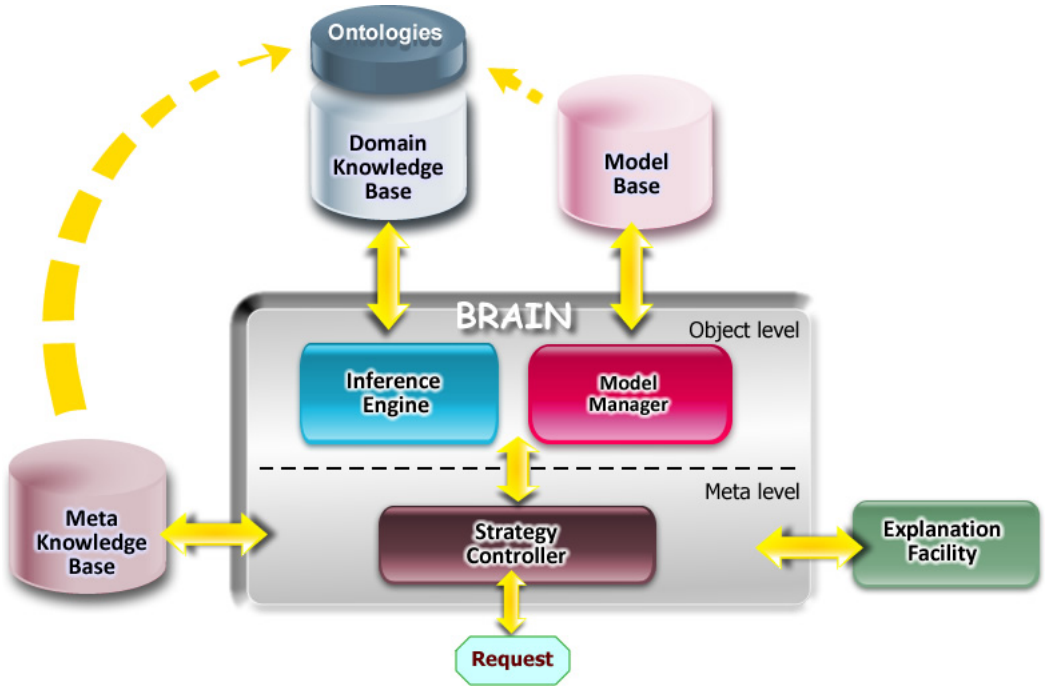


Figure 7-5. *The general view of the HEARTFAID CDSS architecture – dashed arrows correspond to reference to the ontologies, while the others denote a direct communication*

- ***Domain Knowledge Base*** (DKB), consisting of the domain knowledge, formalized from the guidelines and of the clinicians' know-how;
- ***Model Base*** (MB), containing the computational decision models, signals and images processing methods and pattern searching procedures;
- ***Meta Knowledge Base*** (MKB)**,** composed by the strategy knowledge, i.e. about the organization of the CDSS tasks.

[6] As explained in the next section, codifying it in a suitable format for the knowledge base would require too much space. Moreover, considering an external repository improves data, knowledge and system portability.

- ***Brain***, the system component endowed with the reasoning capability, which is divided into the meta and object level;
- ***Explanation Facility***, providing a means to explain conclusions taken.

In particular, the DKB and MKB, which represent the reasoning knowledge of the system, are used by the object level, namely by the *Inference Engine* (IE) and the *Model Manager* (MM), respectively, for inferential and computational reasoning. The *domain level*, introduced in the previous section, is completely represented by a portion of the ontologies, which, however, compose also the DKB. The MKB constitutes the knowledge strategy level and is controlled by the *Strategy Controller*, which is responsible for task accomplishment. The Explanatory Facility

It should be noticed from Figure 7-5, that the ontologies – at least the portion defining the domain structure – are referred (dashed arrows) by both the MB and MKB. In this way, all the data, travelling both inside the CDSS and in the HFP, can be ontologically *lifted*, assuring a higher level of standardization, and reuse possibilities.

Before detailing each component, a brief description of how the system works can be useful. When a request is committed, the meta level component, i.e. the SC, catches it and organises the task for supplying its response. This entails searching, more exactly querying the MKB for the strategy that best fits the actual task. Once found the strategy, being it a simple action or a sequence of subtasks, the SC is responsible of its accomplishment, by controlling and supporting the activities of the object level components.

Considering the case of prognostic evaluation, request R4 in Table 7.I both inferential and computational reasoning are entailed. Hence, the SC strategy would consist in activating both the object components, by forwarding them the necessary data; then wait for their results, combine them (according to a suitable combination mechanism) and supply the final response. Once activated, the IE performs its sub-task by reasons on the actual data and the specific *sub-domain* knowledge of the DKB, i.e. related to prognosis, inferring new facts about the patients. If during the inference process, additional patient's data are necessary, the IE should request their provision to the SC, which has the responsibility to solve such kind of *crisis*, since the only component interacting with the external environment.

The MM applies the computational decision model corresponding to prognosis, assessing the prognostic class the patient belongs to (e.g. *stable*, *improving*, *rapidly worsening*, or *slowly worsening*) and provides the result to the SC.

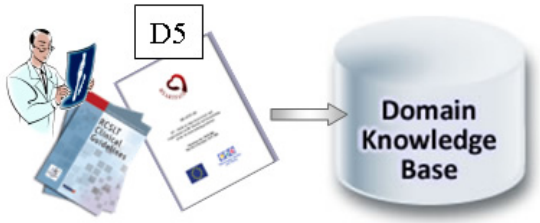
If coded in the strategy, after supplied the prognostic result, the SC can start a new inferential process of the IE for finding out suggestion on the successive clinicians' actions.

The proposed architecture assured several advantages, which fulfil the system requirements, specifying in chapter 6. First of all, the careful separation of processing and knowledge level, both for strategy and reasoning level, assures the effective scalability, and adaptability to changes and growth of the system.

Actually, any addition, deletion or adjustment of the knowledge and model bases (i.e. DKB, MKB and MB) do not affect the processing level of the system. High modularity, careful task decomposition and flexibility are also assured by the separation of the different types of processing, i.e. inference, computational and strategic. On the other hand, the combination of all these different processing capabilities represents an innovative feature that improves the value of the entire platform.
Finally, the virtualization of the CDSS functionalities assures a high degree of portability, fostered also by the system multilevel structure, i.e. only the strategic level and the ontologies are in contact with the external environment.

7.4.1 The Domain Knowledge Base

The Domain Knowledge Base consists in the formalization of the HF domain knowledge, extracted from the clinical guidelines and the physicians' expertise. As already introduced, the combination of ontologies and rules has been selected as the most feasible for the system objectives. A suitable separation of the ontologies according to the reasoning sub-domain has been suggested to reduce the complexity of the system. In order to supply the CDSS functionalities, the DKB should satisfy of set of advisable properties: encoding uncertainty, managing missing parameter, supporting explanation and check for consistency (Figure 7-6).



Content	Structure	Functionalities
• Domain knowledge related to diagnostic, prognostic and monitoring processes	• Ontologies (*sub-ontology* for domain problems) • Rules	• Encode uncertainty • Manage missing parameters • Support explanation • Check of consistency

Figure 7-6. *Summary of the key properties of the Domain Knowledge Base*

7.4.2 The Model Base

The Model Base contains the set of analytical models or procedures developed for supporting decision making.
Functionally, such procedures can be divided into three classes:

- algorithms developed for signals and images processing;
- decision models for computational reasoning;
- pattern recognition algorithms for similarity search of historical data.

Two modelling approaches can be followed for MB modelling: it can be developed as a *library* of procedures, obviously organized on the base of their functionalities; or as a *structured* archive, with a detailed catalogue for algorithms retrieval and application. The latter choice is, of course, more complicated, but assures higher level of flexibility and reusability.
An advisable property of these models is the ability of associating confidence values to their results, which allows assessing their reliability.
Figure 7-7 shows a summary of the MB peculiatities.

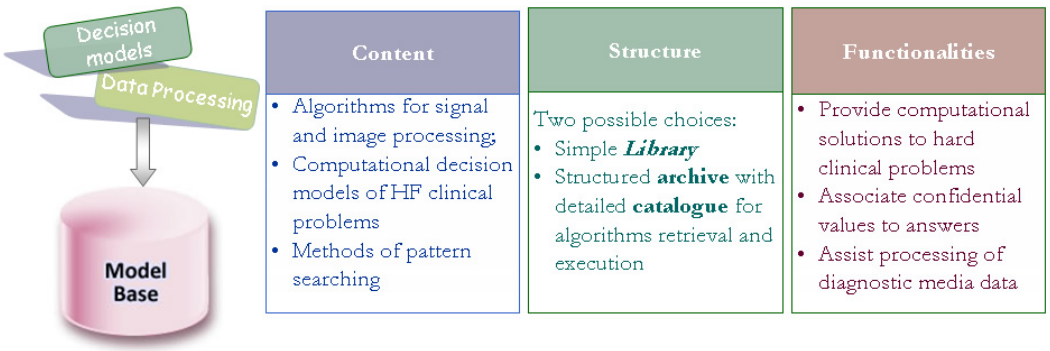


Figure 7-7. *Summary of the key properties of the Model Base*

7.4.3 The Meta Knowledge Base

The Meta Knowledge Base contains the strategic knowledge of the system, i.e. the strategies to be followed for performing tasks. Since consisting mainly in procedural knowledge, they can be represented in rules-based form, and then grouped for simulating a flow activity. Otherwise, more suitable formalisms, able to model activities and workflows, might be adopted.
The procedural typology of this knowledge base can be exploited for coding herein also the guidelines protocols required for satisfying specific CDSS functionalities (e.g. NYHA classification). Moving this kind of domain knowledge to the strategy level is particularly suitable, since protocols compliance requires frequents interactions with the users.

7.4.4 The Brain

As introduced, the brain is the core component of the CDSS, since responsible of its overall decision support activities. For improving its reliability, a two level model has been adopted, separating the meta from the object level.
The high level component, the Strategy Controller, is responsible of the functional behaviour of the underlying object level components, i.e. the Inference Engine and the Model Manager.
In the following each of them are described in more detail.

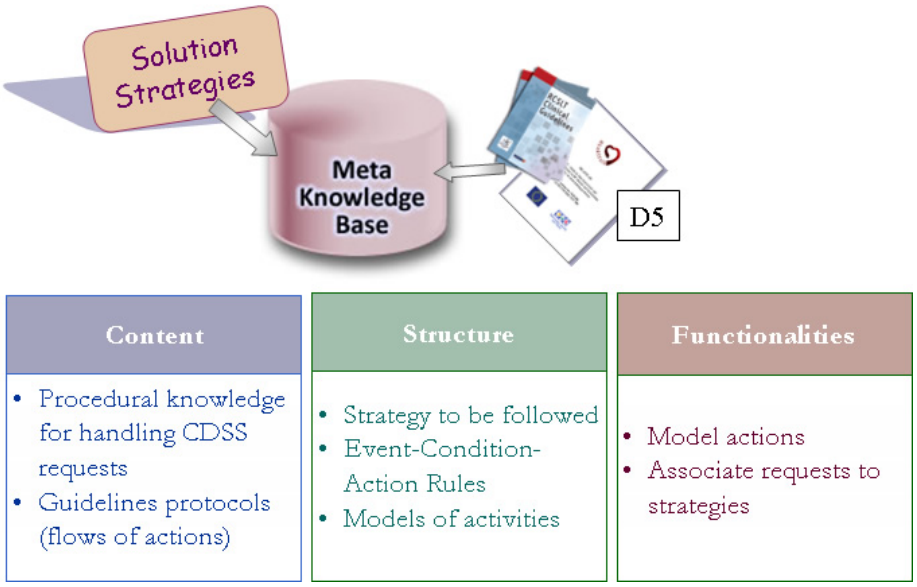


Figure 7-8. *Summary of the key properties of the Meta Knowledge Base*

7.4.4.1 The Strategy Controller

The Strategy Controller handles, organizes and follows the system tasks according to the strategies formalized in the MKB. Its main functionalities consist of:

- Handling the incoming requests;
- Inferring from the MKB the strategy to be followed;
- Applying the strategy, by
  - selecting the suitable object component to activate;
  - managing possible object crisis, such as missing data.

According to what discussed for the MKB, another functionality consists in managing the compliance of guidelines protocols.
The modular structure is composed by:

- a *Request Handler*, which extracts strategies from the MKB,
- a *Strategy Executor*, responsible of applying the strategy,
- a *Crisis Solver*, which provides the solution to critical situations, e.g. uploading missing data from the repository, or forwarding requests to the users.

Figure 7-9 summarises all the described functionalities, shows the internal modular structure of SC and reports its overall functioning.

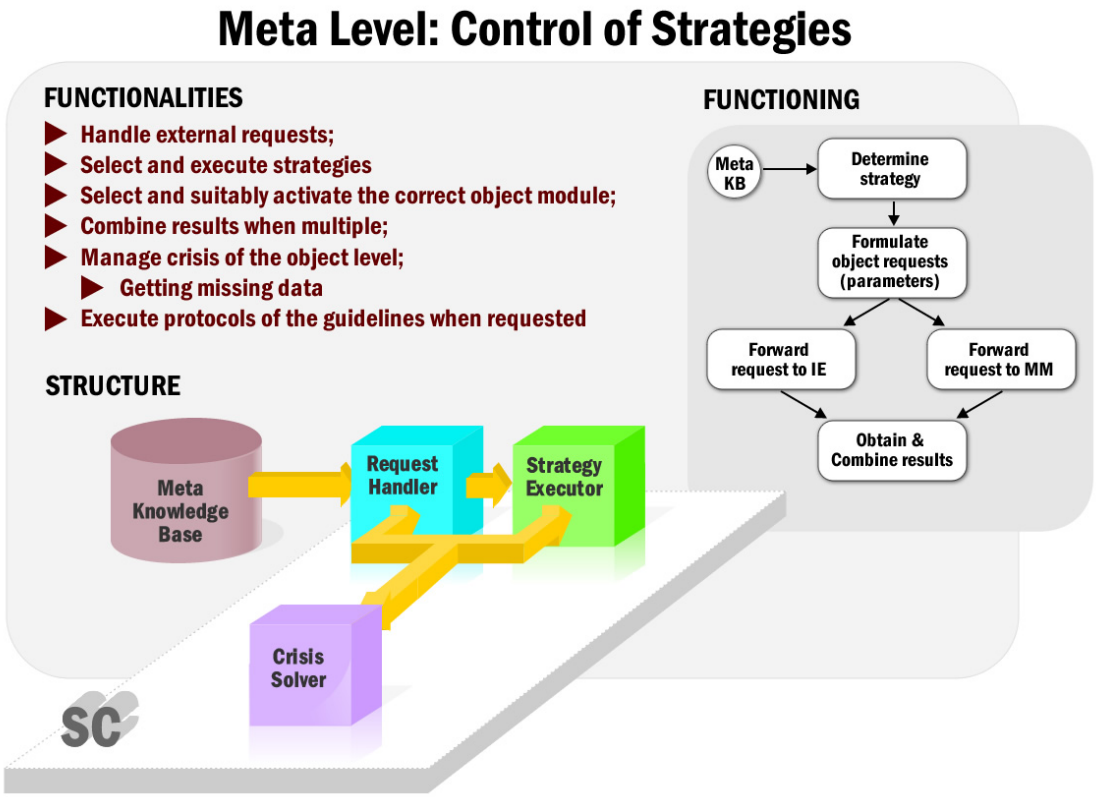


Figure 7-9. *Description of the Strategy Controller in terms of its functionalities, structure and overall functioning.*

7.4.4.2 The Inference Engine

The Inference Engine is devoted to the system inferential reasoning on the symbolic knowledge contained in the DKB. Its main components are

- a *Pattern Matcher*, which finds out the rules to fire and applies them;
- a *Scheduler*, which maintains and order the list of the rules to fire, according to a specific priorities *agenda*;
- a *Working Memory*, which stores the facts inferred while reasoning.

In order to support the CDSS functionalities, the IE should satisfy the following properties:

- allow both forward and backward reasoning, for both inductive reasoning and hypothesis testing (e.g. diagnosis hypothesis of R1 in Table 7.I);
- manage and resolve rules conflicts;
- handle uncertainty;
- handle exceptions.

In particular, conflict resolution and uncertainty handling are tasks of the scheduler.
Figure 7-10 summarises all the described functionalities, shows the internal modular structure of IE and reports an example of its functioning modalities, i.e. forward chaining.

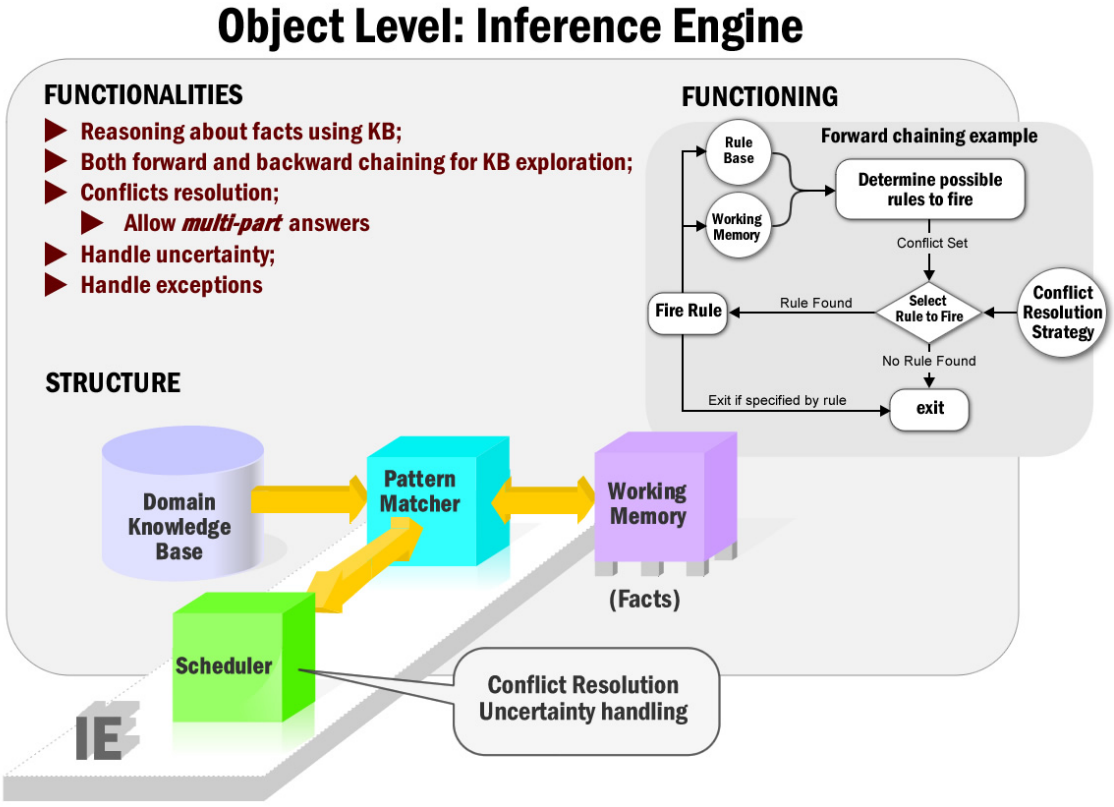


Figure 7-10. *Description of the Inference Engine in terms of its functionalities, structure and functioning examples.*

7.4.4.3 The Model Manager

The Model Manager is responsible of applying the analytical models contained in the MB. Since the latter can be structured according to two different models, the same MM can have different functionalities.
More precisely, if the MB consists in algorithms libraries, the MM performs only call to such libraries. Despite, more complex functionalities are required if the MB is modelled as a structured archive: the MM has to identify the most suitable models and apply it, in some case it can also require the application of more models and then combine their results. The main components of the MM are

- a *Finder*, devoted to models retrieval;
- a *Runner*, responsible of their application;
- a *Combiner*, which combine result according to a specific combination mechanism.

Figure 7-11 summarises all the described functionalities, shows the internal modular structure of MM and reports its overall functioning.

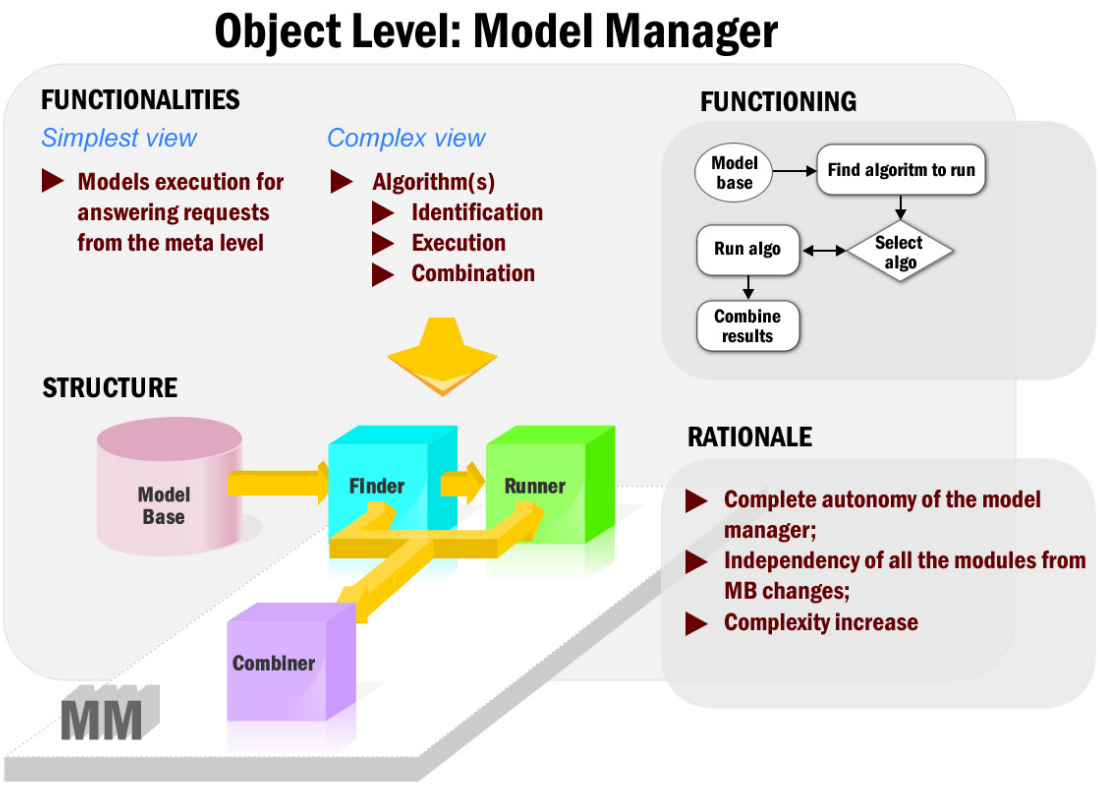


Figure 7-11. *Description of the Model Manager in terms of its functionalities, structure and overall functioning.*

7.4.5 The Explanation Facility

The Explanation Facility is responsible of supplying explanation of the suggestions provided by the system. The explanation can range from how the final or intermediate solutions were arrived at to justifying the need for additional data.
Such functionalities can be fulfilled in different ways: by tracing the fired rules during inference; by adding explanatory comments to the same rules, or by translating simple rule in natural language.
It is connected with the meta level since it can also explain the rationale behind the way in which the system carries out a task in a vocabulary understandable for humans. This makes such a model an important vehicle for communicating about the system both during development and during system execution.

7.5 Design considerations

Looking at the general view of HEARTFAID CDSS architecture (Figure 7-5) we have to consider that the CDSS has to interact with other modules of the HEARTFAID platform: in particular with the HIS Repository (see Deliverable D11), the End-User and a scheduler.

Thinking about data we have considered that the HEARTFAID platform has at least two different models of repositories that can be distributed on the network: the HIS Repository and the Knowledge Base Repository.
The first one is relational, the second one ontological.
Even if the latter could rely on a relational database the upper view of the data structure is different and it is hardly usable by non ontological applications that should have each way to transform data in a different format.
This means that we have duplicate information (in particular, but not uniquely, the patient data is partially stored both in the HIS and in the KB repository).
The KB Repository has to be initialized with ontologies and rules, but also with some instances: in particular the latter could be the information that are valid for any patient, such as information about diagnosis, prognosis, therapies, ...

We have then considered in which ways patient information could be sent to the Ontological database (ODB) in order to answer the question "Who/What instantiates the instance of the patient Vito Gattuso ontology and also in which way.
Four ways are possible:

- All known information about all Patients is sent to the ODB at initialization and each time new data arrive to the HIS repository is also sent to the ODB
- All Patient information is sent to the ODB with the request
- The CDSS asks all information of a Patient after the request
- Part of information is sent with the request, part can be asked by the CDSS (Patient Information needed in the request are sent, other needed by the inference are asked by the CDSS)

Even if the latter and the first one seems the most convenient the latter should be the most appropriate given that the other require an overhead of information to be sent through the network that could cause a bottleneck, a slowdown of all the HEARTFAID module processes and a long suspension of the inference reasoning that should have to wait for missing information.

This leads to specify tasks of the two levels of the HEARTFAID CDSS: Meta level is devoted to control-flow while Object to data-flow.
Specific tasks Meta-level has to perform are:

- to select the appropriate object level
- to fire the appropriate rule (rule-set) when the inference engine is selected
- to resolve crisis (inference conflicts, missing data needed to infer new facts, ...)
- to send inference results to the appropriate HEARTFAID platform module
- to guide end-user interactions (for example in order to choose the appropriate NYHA classification, ...

Specific tasks of the inference object level are:

- to infer new facts

- to ask meta to get missing information

Many inferencing rule languages are extensible with builtin methods (http://www.w3.org/Submission/SWRL/#8): a builtin could be written in order the object can ask the Meta to get missing information.

Thinking about queries we have identified at least two ways of asking questions to the HEARTFAID CDSS:

8. return if is it true that with situation A the conclusion is B, for example as in answering the query “Has the Patient Vito Gattuso heart failure risk ?”.
9. return all your knowledge about a specific item, for example answering a query like as “Patient Vito Gattuso has symptoms A and B: what about the diagnosis?”

Let’s see main IT issues of the first one.
Supposing a physician needs the HEARTFAID CDSS for a diagnosis suggestion, given symptoms of a patient.
The CDSS Meta level selects the inference object level, then selects the diagnosis rule-set passing in input the Patient identification and symptoms A and B.
The inference engine finds four rules, in order:

1] IF symptoms A & B & C THEN Diagnosis D1 40%
2] IF symptoms A & B & D THEN Diagnosis D2 40%
3] IF (Symptoms A & B & C) & (Sign E) & Test F:measure1 >90 THEN Diagnosis D3 98%
4] IF (Symptoms A & B & D) & (Sign F) & Test G:measure2 >0.5 THEN Diagnosis D4 95%

after the first two rules the inference engine suspends the inference because symptom C is missing and asks the Meta to get it (for shortness of description, in reality this should happen each time a rule miss an information).
The Meta level asks for symptoms C and D to the HIS Repository and the latter answers returning the requested information: C and D are both true.
The first two rules are both true but also rules 3 and 4 are fired.
The inference fires the other two rules but sign E and F, measure 1 of Test F and measure 2 of Test G are missing information so it asks the Meta to get these.
The Meta level asks signs E and F to the physician that enters them using an user interface: E and F are both true.
Then Meta asks the physician to apply protocol for test F in order to get measure 1
Physician performs protocol for test F and provides measure 1 to the CDSS: Measure1 = 50
The third rule results false.
Then the Meta asks the physician to apply protocol for test G in order to get measure 2
Physician performs protocol for test G and provides measure 2 to the Brain Meta Level: Measure2 = 0,7

Rule four results true, then the inference engine returns as result the conclusion that Diagnosis D4 is true at 95%.

Going down in detail, even the certainty of single part of the premises of a rule would be not evaluated with classic but with fuzzy logic: for example symptoms and signs could not have a clear-cut division between true and false but they would have certainty degrees.

The second case is simpler; the main IT issues are included in the first case, apart that the backward reasoning is use instead.

From the IT point of view
The CDSS has to take into account not only inferences on declarative information but also interactions with other HEARTFAID platform modules, procedural steps, for example:

- Asks the HIS Repository for information
- Sends info about therapies and exams to the Scheduler
- Activates alarm module ALERT
  IF problem_severity > high_severity THEN alert physician

And has also to guide through pathway steps physicians and patients have to perform, for example
IF patient has problem 3 THEN Pathway X should to be followed
where Pathway X can be constituted by the following steps
Patient can come back to home,
Patient has to assume Therapy 7,
After 15 days exam 8 is required,
After exam 8 a new diagnosis is required.

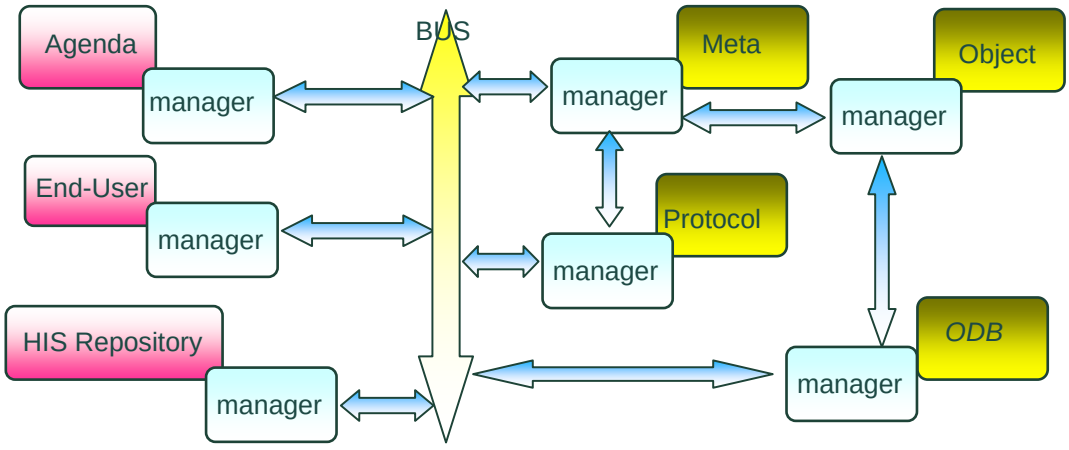


Figure 7.12. *Information flows involving HEARTFAID platform modules and the HEARTFAID CDSS*

Considering Figure 7.12, representing flow occurring among main modules involved in the decision process, we have to point out that each module has to have a manager able to transform information between request and response and whenever a module is found in the flow a format transformation is required by its manager.

Some standards the HEARTFAID CDSS should have to work with have been analysed by WP4, in particular OWL. SPARQL is usable in conjunction given its matureness and similarity with the Standard Query Language (SQL) of relational databases.
About rules, Rule Interchange Format is not a standard yet.
Hoping it will become cause the main issues the RIF Group is facing are recognized as topic problems a look at its compliance should be given by developers of the HEARTFAID CDSS.
To be compliant with RIF means that the rule language should at least have a XML syntax, support
XML Schema Datatypes (XSD).

An interesting case reported by the RIF Group is the RIF UseCase 2.6:
“Ruleset Integration for Medical Decision Support”.
In this case is highlighted the needs of using a variety of sources in the reasoning process, in particular three sources are specified:

- a pharmaceutical knowledge base
- a patient databases, which gives the patient record, including the medications a patient is currently taking
- a hospital medical event protocol knowledge base, which details the protocol used for different medical procedures

This is only a use, a solution hasn't already proposed.
In our case, a homogenization of the sources appears the most appropriate, in order to merge the KB without interchange problems. A conversion of data coming from the HIS Repository is anyway needed as examined previously.

8. Towards a HF CDSS Implementation

In this Section we explore possible approaches and solutions to CDSS implementation.
As described in Chapter 4, there are many instruments and tools that can be considered useful for possible implementations.
Of course all the constraints that we have pointed out in some previous sections of this document shall be taken into account in order to define an up-to-date 'universal' approach, able to characterize a solution where standards, efficiency and robustness are key factors.
In order to better explain how these tools can be employed to develop the CDSS, in the following we investigate a certain case as a paradigm.
In particular, we put attention to selected tools which could be used to implement all the components that compose the whole computational chain of our system.

- *Create an Ontology*

Let us consider two editors: Protegè and Swoop.
The first one (in Figure 8.1) is probably one of the most known and used editors, well cared also in the graphical aspect; it was developed by the Stanford Medical Informatics at the Stanford University School of Medicine. In general, we used it successfully in many applications to create RDF files and OWL ontologies.

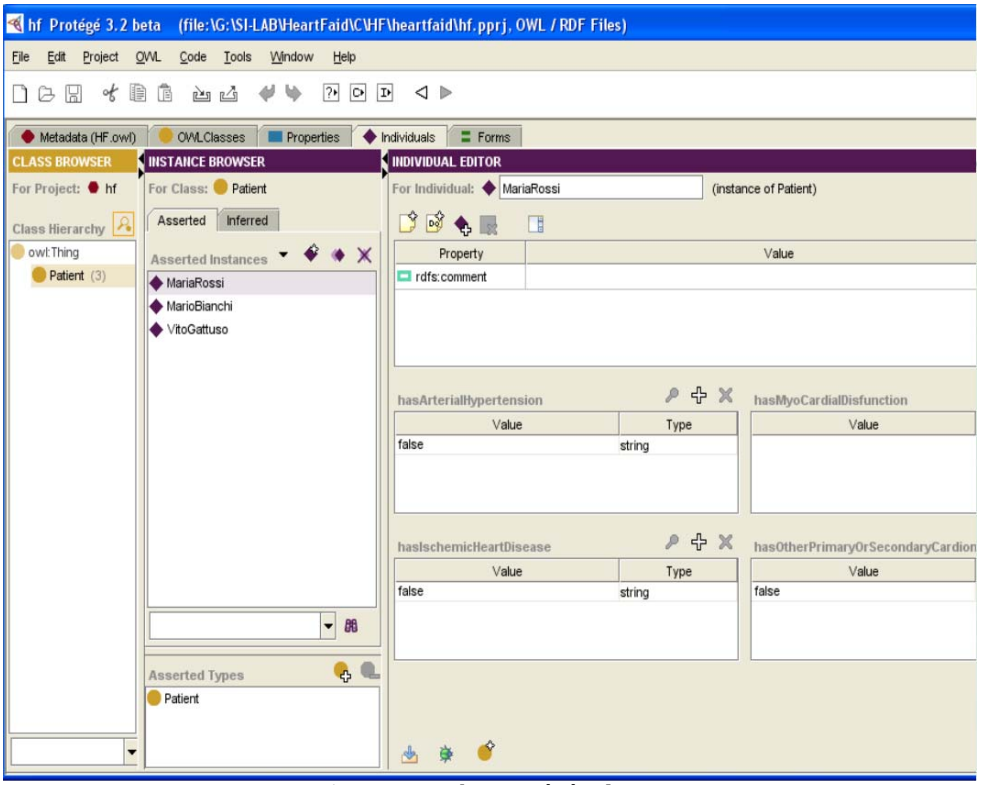


Figure 8.1. *The Protégé Editor.*

However, we found problems in merging complex ontologies: for instance, it is not able to manage MPEG-7 together with other OWL ontologies.

Swoop, initially (in Figure 8.2) developed by a small team of well known experts of Semantic Web technologies, has currently been adopted by the Google Code Initiative: it is lighter and offers fewer features than Protegè but it is very robust (and successfully loads the over said ontologies).

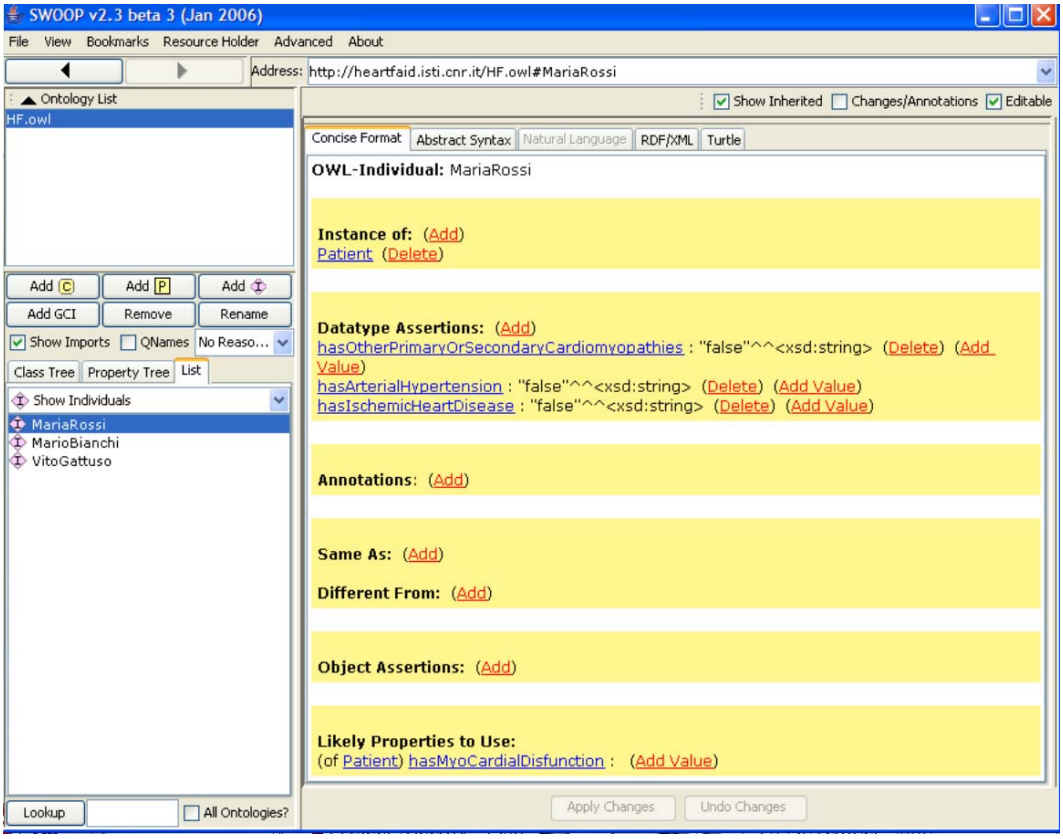


Figure 8.2. *The Swoop Editor.*

Both the tools provide the possibility to edit and also to check ontologies given their connectivity to an inference engine.
Protegè also offers a tool to write SWRL rules and this can be considered a quite positive aspect since not many open-source or free editors of rules exist.
Nevertheless, these editors are probably too complex for being used or even considered by physicians: a more friendly approach could be then realising a graphical user interface providing a visual and textual help to a doctor who needs to add or modify directly the inferencing rules.

- *Use an Inference Engine*

Once created the Knowledge Base (RDF, OWL, and rules), Jena could be considered as a solution to implement our inference engine.
Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine.
Probably Jena represents the most widely adopted solution, because it is an open-source software and it is a robust and mature product. It was developed by Hewlett Packard and it has a great number of active supporters.
Other interesting tools are also Mandarax and Jess.
The first one uses RDF, OWL and RuleML as standards. Unfortunately, it is not more supported from 2004.
Jess, instead, which is another good tool currently at its 7th release (most part of these products doesn't arrive to the 2nd one), is unfortunately a commercial product.
Jena rule-based reasoners provide the possibility to use RDF, OWL (Lite, some constructs from OWL-DL and OWL-full). Concerning large or complex ontologies, developers suggest using Jena in conjunction with an external reasoner implementing the DIG description logic reasoner interface [DIG Interface]. This interface is an emerging standard for providing access to description-logic reasoning via an HTTP-based interface to a separate reasoning process (similarly to the ODBC in relation to databases).
In particular Jena's developers suggest using their product in conjunction with the Pellet open source reasoner.

We have performed some preliminary tests both with Pellet and Bossam having good results: they support OWL and the SWRL rules.

The standard proposed by W3C in order to query an ontology is SPARQL.
Jena provides the possibility to query ontologies using SPARQL with the ARQ query engine.
Also Pellet provides a SPARQL query engine.
Bossam does not support SPARQL but it provides the Buchingae query language. Buchingae is a simple web-oriented rule language. By "web-oriented", we mean that it is possible to refer directly to URI resources, such as web ontology elements or RDF resources, when writing rules. Buchingae is a language appropriate for specifying production rules, but it is not limited to production rules as it has procedural attachment features that enable writing ECA rules. This is an interesting feature like the *builtin* extension provided by the Jena rules.
We haven't seen this opportunity with the current stable Pellet version.

- *Realize a Strategy Controller*

The strategy controller could be realized in several ways:

- ad-hoc
- using a workflow engine
- using rule extensions (rule builtins)
- mixed

The first one foresees a custom implementation based on the use of a programming language (e.g., Java).
The second one, instead, supposes that the controller is built using an external workflow engine which is in its turn based on procedural rules.
The third one supposes that the inferencing rules must be extended in order to provide also procedural rules.
Finally, the mixed solution could be used also thinking at the pathways/protocols that the CDSS has to handle in order to guide doctors in their activities.

The first one is not sufficiently flexible.
To adopt the second or the third solution, developers should consider how rules are written: workflow engine may be more efficient to retrieve sets of missing information before the inference engine is launched, whilst the rule extensions may be simpler to realize.
A good compromise would be the use of a mixed approach.

Depending on the variability of the procedural flow inside the HEARTFAID platform, we think that the workflow solution could be used to manage all the events (see D11): note that this functionality is not a part of the CDSS.

- *Map a HIS Repository to a KB*

In order to bridge the gap between the relational schemata used by the HIS Repository and the ontologies of the KB, we could follow the way to test a DB-KB Mapper.
This kind of tool exports data and maps queries between the two above mentioned data and metadata structures.
D2Rmap and DartGrid are interesting examples of such kind of products. [Martin O'Connor 2007] [D2R MAP] [DartGrid].

Preliminary results show interesting capabilities even for those products that do not reach a stable version yet.

9. Conclusions

Activities focused on two main topics, namely:

- Data processing (Task 5.1, 5.2);
- Decision Support Services (Task 5.3).

About *Data Processing*, a state of the art was carried out regarding the representation and the characterisation of the relevant features extracted from signals and images, both in their general properties and in the specific cardiovascular field under investigation.
Preliminarily, it was necessary to acquire a schematic though precise knowledge of the medical domain. With this goal in mind, a careful examination of the medical-clinical needs, the guidelines by the ESC and ACA/AHA and of related literature cited therein was performed.
In this way, it was possible to draft the lists of both the diagnostic devices and the representation features relevant for the various levels of the HEARTFAID platform. Some of the considered features had a precise mathematical foundation, while other ones seemed to be only of a qualitative nature.
Besides, two main aspects when interpreting cardiologic signals and images were considered: open problems assessment and correlation analysis and signal/image categorization. The first one, referred to intra/inter-observer variability, was studied to reduce instrumentation lacks and analysis subjectivity. The latter instead aimed at extending the HEARTFAID knowledge through developing new methods for extracting *innovative representing features.*
Once identified the classical cardiac parameters which are more prone to intra- and inter-observer variability, we studied algorithms for left ventricle end-systolic and end-diastolic volume quantification in ultrasound images. During the last month, a testing activity started on left ventricle segmentation from echocardiographic image sequences (4 and 2 chambers apical views and short axis mid-papillary) (Task 5.2).
In relation to innovative representing features, we started developing and testing tools for the analysis and fast visualization of segmental timing mechanics. This way, we intended to assess the cardiac dyssynchrony in heart failure patients and to foresee the outcome of cardiac resynchronization therapy. Our goal was also to extract innovative representing features to improve analysis by adding input data to the HEARTFAID CDSS, useful to categorize the left ventricle deformation pattern.

The work related to *Decision Support Services* started with a deep investigation of the methodological foundations of Decision Support Systems (DSS), both in organizational and clinical applications. In so doing, the general requirements and functionalities of a DSS were listed up. Particular attention was reserved to knowledge based clinical systems, and guidelines modelling approaches.
In order to map the studied methodologies into the HEARTFAID domain, we firstly analyzed the platform as a complex clinical information management

system, able to offer a number of services for optimizing the daily clinical practice. In this perspective, the HEARTFAID *cycle* was outlined as consisting in *measurements*, *analysis*, *decisions* and *actions* phases.
Once the several services to be supplied were carefully investigated, three main contexts were identified, i.e. data collection and management, knowledge-based decision support and end-user applications, assigning them some of the platform functionalities. Then, we focused on the knowledge-based decision support context, i.e. the HEARTFAID CDSS. An accurate investigation was then performed concerning (i) the medical-clinical requirements and the relevant decision making problems, (ii) the users' scenarios, and (iii) the project requirements. This led to the definition of the overall characteristics of the HEARTFAID CDSS.
In particular, four main application scenarios were pointed out, that is heart failure diagnosis, prognosis, therapy planning and follow up. For each of them, more detailed clinical problems were translated into requirements of the CDSS. In addition, important success factors were also highlighted: completeness on the important *issues*, and ability to show and let the users explore the whole *space* of possibilities; representation aiding; capability of explaining the suggested decisions; adaptability to changes, also in accordance with the end-users applications, and ability to check the correctness and consistency of the knowledge base; capability of dealing with uncertainty. Finally, other important system characteristics related to the system implementation were stated, e.g., robustness, computational optimization, portability, open-source, and so on. The important question of system proactivity vs. reactivity was also considered and the achievement of both features evaluated.
Starting from the requirements, the functionalities of the CDSS, considered as a resource of the platform activated *on-demand*, were determined.
As a result, a logical architecture of the system was then outlined. The following core components were identified:

- the knowledge base – consisting of ontologies and rules, and from the domain knowledge conceptualization process;
- the model base – containing all the *computational* decision models, included the ones extracted from the KDD processes, and the algorithms for data processing;
- the *Brain* – i.e. the intelligent part of the system - able to manage and support the high level decision requests. The *Brain*, designed to exploit the knowledge and the model bases, was structured in two parts, *meta* and *object*, the former acting as a strategy controller on the basis of a specific protocol, and the latter divided into two logical parts (i) an inference engine (able to reason on selected sub-domains), and (ii) a *model manager* (able to handle models and their applications).

In particular, to assure a correct integration, the benefit of ontologically lifting the different components was underlined.

This way, we proposed to provide a reference to a *central* ontology for assigning a unique semantics to common and shared information. Thus, easiness of communications, knowledge improvements and methods standardization and reusability were assured.

Furthermore, an extensive investigation of the available technologies was carried out, with emphasis on robust, portable and open-source instruments.
Guidelines for the CDSS implementation and for its interactions with the other components of the HEARTFAID platform were also defined.

All the above issues have been deeply discussed and explained in this document.

References and main Bibliography

Alter, 1980	Alter, S. L.. "Decision support systems: current practice and continuing challenges". Reading, Mass., Addison-Wesley Pub.
Amann *et al.* 2005	Amann A, Tratnig R, and Unterkofler K. "Reliability of old and new ventricular fibrillation detection algorithms for automated external defibrillators". Biomed. Eng. Online, No. 4, Vol. 60, 2005
Amini *et al.* 1998	A.A. Amini, Y. Chen, R.W. Curwen, V. Mani, and J. Su, *"Coupled B-Snake grids and constrained thin plate splines for analysis of 2-D tissue deformations from tagged MRI", IEEE Trans. Med. Imag*.17, no. 3, 344-356, 1998
Aristotle 300B.C.	Aristotle, *Metaphysica*, Werner Jaeger ed., Oxford Classical Texts. Oxford University Press, 1957
Asirelli *et al.* 2006	Asirelli P., Little S., Martinelli M., Salvetti O., "MultiMedia metadata management: a proposal for an infrastructure", *Semantic Web Applications and Perspectives. SWAP 2006 (Pisa, 18-20 December 2006)*, Proceedings online. G. Tummarello, P. Bouquet, O. Signore (eds.). CEUR, 2006
ATS 1995	American Thoracic Society. Medical section of the American lung association. "Standardization of spirometry. 1994 Update". American Journal of Respiratory and Critical Care Medicine, Vol. 152, pp. 1107-1136, 1995
ATS 2002	American Thoracic Society. "ATS statement: guidelines for the six-minute walk test". American Journal of Respiratory and Critical Care Medicine, Vol. 166, pp. 111-117, 2002
Bax *et al.* 2004	Bax JJ, Bleeker GB, Marwick TH, Molhoek SG, Boersma E, Steendijk P, *et al*. "Left ventricular dyssynchrony predicts response and prognosis after cardiac resynchronization therapy." *J Am Coll Cardiol*, vol. 44, 1834-40, 2004
Bemmel 1982	Van Bemmel JH. "Recognition of electrocardiographic patterns". In: *Handbook of statistics*, Krishnaiah PR and Kanal LN (eds), Amsterdam North Holland, 1982
Binder *et al.* 1999	T. Binder, M. Sussner, D. Moertl, H. Strohmer, T. Baumgarten, G. Maurer, G. Porenta, "Artificial neural networks and spatial temporal contour linking for automated endocardial contour detection on echocardiograms: A novel approach to determine left ventricular contractile function", *Ultrasound Med. Biol.*, 25, pp. 1069, 1076, 1999
Bosch *et al.* 2002	J.G. Bosch, S.C. Mitchell, B.P.F. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, J.H.C. Reiber, "Automatic segmentation of echocardiographic sequences by active appearance motion models", *IEEE Trans. Med. Imag.*, 21, pp. 1374-1383, 2002
Brachman, 1979	Brachman R.J. "On the epistemological status of semantic networks". In N.V. Findler, editor, *Associative Networks: The Representation and Use in Computers*, pp 3-50. New York: Academic Press Inc, 1979
Brown *et al.* 1995	M. Brown, R. Gill, H. Talhami, L. Wilson, and B. Doust. "Model-based assessment of lung structures: Inferencing and control system". In: *Proc. SPIE Conf. Medical Imaging*, vol 2433, 167-178, 1995
Brown *et al.* 2007	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, last retrieved in February 2007)
Caceres 1963	Caceres, C. A., "Electrocardiographic analysis by a computer system", *Arch Intern Med*, 111, pp. 196–202, 1963

Chakraborti and Duncan 1999	Chakraborti, J. Duncan, "Game-theoretic integration for image segmentation", *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 21, pp. 12-30, 1999
Chalana *et al.* 1996	V. Chalana, D.T. Linker, D.R. Haynor, Y. Kim, "A multiple active contour model for cardiac boundary detection on echocardiographic sequences", *IEEE Trans. Med. Imag.*, 15, pp. 290-298, 1996.
Chandra *et al.* 2003	Chandra T, Yeates DB, and Wong LB. "Heart Rate Variability Analysis – Current and Future Trends". Business Briefing: *Global Healthcare,* 2003
Chandrasekaran *et al*, 1999	Chandrasekaran R., Josephson J., Benjamins V."What are ontologies, and why do we need them?" *IEEE Intelligent Systems*, 14(1):20–26, January/February 1999
Chen, 2000	Chen Z. "Computational intelligence for decision support". CRC Press International series on computational intelligence, 2000
Chen *et al.* 2003	Y. Chen, F. Huang, H.D. Tagare, R. Rao, D. Wilson, E.A. Geiser, "Using prior shape and intensity profile in medical image segmentation", *Proc. IEEE Int. Conf. Comput. Vis.*, Nice, vol. II, pp. 1117-1125, 2003
Chiarugi *et al.* 2001a	Chiarugi F, Lombardi D, Lees PJ, Chronaki CE, Tsiknakis M, and Orphanoudakis SC. "Integration of an ECG management system with an existing cardiology information system to support daily ECG procedures in a cardiology department", presented at the *XXIII Congress of the European Society of Cardiology* (ESC), Stockholm, 1-5 September 2001
Chiarugi *et al.* 2001b	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?", presented at the *28th Annual Computers In Cardiology Congress*, Rotterdam, The Netherlands, 23-26 September 2001
Chiarugi *et al.* 2002	Chiarugi F, Lombardi D, Lees PJ, Chronaki CE, Tsiknakis M, and Orphanoudakis SC. "Support of Daily ECG Procedures in a Cardiology Department via the Integration of an Existing Clinical Database and a Commercial ECG Management System", *Annals of Noninvasive Electrocardiology*, vol. 7 (2), 2002
Chiarugi *et al.* 2003	Chiarugi F, Spanakis M, Lees PJ, Chronaki CE, Tsiknakis M, and Orphanoudakis SC. "ECG in Your Hands: A Multi-Vendor ECG Viewer for Personal Digital Assistants", presented at *Computers in Cardiology*, 2003
Chronaki *et al.* 2003	Chronaki CE, Chiarugi F, Lees PJ, Charalambus E, Sykianakis A, Vrouchos G, Antonakis N, Frantzeskakis G, Tsiknakis M, and Orphanoudakis SC. "Digital ECG Databases in the Regional Health Information Network of Crete", presented at *ICE2003, XXX International Congress on Electrocardiology* 2003
Cleland *et al.* 2005	J.G. Cleland, J.C. Daubert, E. Erdmann *et al*. "The effect of cardiacd resynchronization on morbidity and mortality in heart failure. *N. Engl. J. of Med*, vol. 352, 1539-49, 2005
Clifford 2002	Clifford GD. *Signal processing methods for heart rate variability analysis*. Academic dissertation at the Department of Engineering Science, University of Oxford, 2002
Colantonio *et al.* 2005a	Colantonio S., Di Bono M.G., Pieri G., Salvetti O., "Processing multimedia biomedical information for disease evolution monitoring", *Ercim News*, 60, pp. 28-29, 2005
Colantonio *et al.* 2005b	Colantonio S., Moroni D., Salvetti O., "A methodological approach to the study of periodically deforming anatomical structures", *Advanced Information and Telemedicine Technologies for Health (Minsk, Belarus, November 8-10, 2005)*, pp. 32-36, 2005

Colantonio *et al.* 2005c	Colantonio S., Di Bono S., Salvetti O., "Disease evolution prognosis based on multi-source signals and image analysis", *Advanced Information and Telemedicine Technologies for Health (Minsk, Belarus, November 8-10, 2005)*, pp. 66-69, 2005
Colantonio *et al.* 2005d	Colantonio S., Moroni D., Salvetti O., "MRI Left ventricle segmentation and reconstruction for the study of the heart dynamics", *IEEE 5th Symposium on Signal Processing and Information Technology. ISSPIT (Athens, Greece, December 18-21, 2005)*, 2005
Colantonio *et al.* 2006	Colantonio S., Moroni D., Salvetti O., "Shape comparison and deformation analysis in biomedical applications", *4th Eurographics Italian Chapter 2006*, pp. 37-43, 2006
Davis *et al.*, 1993	Davis R., Shrobe H., Szolovits P. "What is a knowledge representation?" *AI Magazine*, 14(1):17–33, 1993
Declerck *et al.* 1997	J. Declerck, J. Feldmar, M.L. Goris, and F. Betting, "Automatic Registration and Alignment on a Template of Cardiac Stress and Rest reoriented SPECT Images", *IEEE Transactions on Medical Imaging*, 16(6), pp. 727-737, 1997
deDombal 1975	deDombal, F. T., "Computer-aided diagnosis and decision-making in the acute abdomen", *J R Coll Physicians Lond*, 9(3), pp. 211–218, 1975
Dohi *et al.* 2006	Dohi K, Pinsky MR, Kanzaki H, Severyn D, Gorcasan J III. "Effects of radial left ventricular dyssynchrony on cardiac performance usinbg quantitative tissue Doppler radial strain imaging". *J Am Soc Echocardiogr*, vol. 19, 475-482, 2006
Donni *et al.* 1998	Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Andrea Schaerf, "AL-log: integrating Datalog and description logics". *J. of Intelligent and Cooperative Information Systems*, 1998
Enright 2003	Enright PL. "The Six-Minute Walk Test". *Respiratory Care*, Vol. 48(8), 2003
ESC-NASPE 1996	Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology. "Heart rate variability – Standards of Measurements, Physiological Interpretation, and Clinical Use". *Circulation.* 93, pp. 1043-1065, 1996
Faber *et al.* 1991	T.L. Faber, E.M. Stokely, R.M. Peshock and J.R. Corbett, "A Model-Based Four-Dimensional Left Ventricular Surface Detector", *IEEE Transactions on Medical Imaging*, 10(3), pp. 321-329, 1991
Fernantez *et al.* 2003	Fernantez AR, Folgueras J, and Colorado O. "Validation of a Set of Algorithms for Ventricular Fibrillation Detection: Experimental Result". *Proceedings of the 25th Annual EMBS International Conference*, Cancun, Mexico, 2003
Fischer *et al.* 2003	Fischer R, Chiarugi F, Schmid JJ, Norgall T, and Zywietz Chr. "Communication and Retrieval of ECG Data: How Many Standards Do We Need?", *Computers in Cardiology*, 30, pp. 21- 24, 2003
Fischer *et al.* 2006	Fischer R, Chiarugi F, and Zywietz TK. "Enhanced Integrated Format and Content Checking for Processing of SCP ECG Records", *Proceedings of Computers in Cardiology*, 2006
Forbes 1995	Forbes AD. "Classification-algorithm evaluation: five performance measures based on confusion matrices". *J Clin Monit*, 11, pp.189-206, 1995
Frangi *et al.* 2001	Frangi A, Niessen WJ, Viergever MA. "Three-dimensional modelling for functional analysis of cardiac images: A review". *IEEE Trans Med Imag*, vol 20, no 1, 2-25, 2001
Franke and Kohl 2003	M. Franke and H.P. Kohl. "Second-generation real-time 3D echocardiography: a revolutionary new technology". *Medicamundi*, 47(2), p. 3340, 2003
Fedorowicz, 2003	Fedorowicz J. "A technology infrastructure for document-based decision support systems". *Decision support systems (3rd ed.): putting theory into practice*, pp. 125 - 136, Prentice-Hall Inc., 2003

Garcia-Vera *et al.* 2004	Garcia-Vera M, Sanz J, and Labrador FJ. "Blood pressure variability and stress management training for essential hypertension". *Behavioral Medicine*, Summer 2004
Gatos *et al.* 2006	B. Gatos, I. Pratikakis, S.J. Perantonis, "Adaptive degraded document image binarization", *Pattern Recognition*, 39, pp. 317-327, 2006.
Giarratano and Ripley, 1998	Giarratano, J. and Riley, G.**,** "Expert Systems: Principles and Programming" (3rd ed.), PWS Publishing, Boston, 1998
Gorcsan *et al.* 2004	Gorcsan J III, Kanzaki H, Bazaz R, Dohi K, Schwartzman D. "Usefulness of echocardiographic tissue synchronization imaging to predict acute response to cardiac resynchronization therapy". *Am J Cardiol*, vol. 93, 1178-81, 2004
Gorry and Barnett 1968	Gorry, G. A. and Barnett, G. O., "Experience with a model of sequential diagnosis", *Comput Biomed Res*, 1(5), pp. 490–507, 1968
Gotardo *et al.* 2006a	Paulo F. U. Gotardo, Kim L. Boyer, Joel Saltz and Subha V. Raman "A New Deformable Model for Boundary Tracking in Cardiac MRI and Its Application to the Detection of Intra-Ventricular Dyssynchrony" *Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, vol. 1, pp. 736-743, New York, June 2006
Gotardo *et al.* 2006b	Paulo F. U. Gotardo, Kim L. Boyer, Joel Saltz and Subha V. Raman. "On the Detection of Intra-Ventricular Dyssynchrony in the Left Ventricle From Routine Cardiac MRI", *Proceedings of the 2006 IEEE Int. Symp.on Biom. Imaging (ISBI 2006)*, vol. 1, pp. 169-172, Arlington, Virginia, April 2006
Greenes 2003	Greenes R.A. - Decision Support at the Point of Care: Challenges in Knowledge Representation, Management, and Patient-specific Access. *Adv Dent Res* 17:69-73, December, 2003
Greenes *et al.* 1989	Greenes, R. A., Tarabar, D. B., Krauss, M., Anderson, G., Wolnik, W. J., Cope, L., *et al*., "Knowledge management as a decision support method: a diagnostic workup strategy application", *Comput Biomed Res*, 22(2), pp. 113–135, 1989
Greenes *et al.* 2001	Greenes, R. A., Peleg, M., Boxwala, A., Tu, S., Patel, V. and Shortliffe, E. H., "Sharable computer-based clinical practice guidelines: rationale, obstacles, approaches, and prospects.", *Medinfo* 10(1), pp. 201–205, 2001
Holsapple and Whinston 1996	Holsapple, C.W., and A. B. Whinston, *Decision Support Systems: A Knowledge-Based Approach*. West Publishing, St. Paul, 1996
Huang *et al.* 2004	X. Huang, D. Metaxas, and T. Chen, "MetaMorphs: Deformable Shape and Texture Models", in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, CVPR'04*, Oral Presentation, Washington, D.C., June, 2004.
Huang *et al.* 2005	Q. Huang, W. Gao, W. Cai, "Thresholding technique with adaptative window selection for uneven lighting image", *Pattern Recognition Letters*, 26, pp. 801-808, 2005
Hung *et al.* 2007	Hung *et al.* "3D Echocardiography: A Review of the Current Status and Future Directions". *J Am Soc Echocardiogr*, 20, pp 213-233, 2007
Hunt *et al.* 2005	S.A. Hunt *et al*., The American College of Cardiology/American Heart Association Task Force on Practice Guidelines (Writing Committee to Update the 2001 Guidelines for the Evaluation and Management of Heart Failure), "ACC/AHA 2005 Guideline update for the diagnosis and management of Chronic Heart Failure in the adult", *ACC/AHA*, 82 pages, 2005
Inza *et al.* 2001	Inza, I., Merino, M., Larranaga, P., Quiroga, J., Sierra, B. and Girala, M., "Feature subset selection by genetic algorithms and estimation of distribution algorithms. A case study in the survival of cirrhotic patients treated with TIPS", *Artif Intell Med*, 23(2), pp. 187–205, 2001
Jekova and Krasteva 2004	Jekova I, and Krasteva V. "Real time detection of ventricular fibrillation and tachycardia". *Physiol. Meas*., Vol. 25, pp. 1167-1178, 2004

Jekova *et al.* 2002	Jekova I, Dushanova J, and Popivanov D. "Method for ventricular fibrillation detection in the external electrocardiogram using nonlinear prediction". *Physiol. Meas.*, Vol. 23, pp. 337-345, 2002
Jenksins 1996	Dean Jenksins, "A (not so) brief history of electrocardiography", 1996 http://www.ecglibrary.com/ecghist.html.
Jiménez *et al.* 2006	Antonio Jiménez, Sixto Ríos-Insua, Alfonso Mateos, "A generic multi-attribute analysis system", Computers and Operations Research, 33(4), pp.1081-1101, April 2006
Jintrawet 1995	A. Jintrawet, "A Decision Support System for Rapid Assessment of Lowland Rice-based Cropping Alternatives in Thailand", *Agricultural Systems*, 47, pp. 245-258, 1995
Jolly *et al.* 2001	M.-P. Jolly, N. Duta, G. Funka-Lea, "Segmentation of the Left Ventricle in Cardiac MR Images", *ICCV 2001*, pp. 501-508, 2001
Kanzaki *et al.* 2006	Kanzaki H, Bazaz R, Schwartzman D, Dohi K, Sade LE, Gorcsan J III. "A mechanism for immediate reduction in mitral regurgitation after cardiac resynchronization therapy: insights from mechanical activation strain mapping. *J Am Coll Cardiol*, vol. 44, 1619-25, 2004
Kass *et al.* 1987	M. Kass, A. Witkin, D. Terzopoulos. "Snakes: active contour models" In: *International Conference on Computer Vision*, London, pp. 259-268, 1987.
Kassirer *et al.* 1987	Kassirer, J. P., Moskowitz, A. J., Lau, J. and Pauker, S. G., "Decision analysis: a progress report", *Ann Intern Med*, 106(2), pp. 275–291, 1987
Kawamoto *et al.* 2005	Kawamoto K., Houlihan C.A., Balas E.A., Lobach D.F. "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success". *BMJ*, 330, pp. 765- 72, 2005
Keen 1978	Keen, P. G. W., *Decision support systems: an organizational perspective*, Addison-Wesley, 1978
Keen 1980	P.G.W. Keen, "Decision support systems: a research perspective", in: *Decision support systems : issues and challenges*, IIASA proceedings series, Fick and R. H. Sprague, Pergamon Press, 1980
Kemp *et al.* 1992	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, pp. 391-393, 1992
Knaup *et al.* 2002	Knaup, P., Wiedemann, T., Bachert, A., Creutzig, U., Haux, R. and Schilling, F., "Efficiency and safety of chemotherapy plans for children: CATIPO—a nationwide approach", *Artif Intell Med*, 24(3), pp. 229–242, 2002
Kohler *et al.* 2002	Kohler BU, Hennig C, and Orglmeister R. "The principles of software QRS detection". *IEEE Engineering in Medicine and Biology*, January/February 2002
Kolovski *et al.* 2006	Vladimir Kolovski, Bijan Parsia and Evren Sirin, " Extending SHOIQ(D) with DL-safe Rules: First Results". *Proceedings of the International Workshop on Description Logic (DL-2006)*, 2006
Krticka *et al.* 2000	Krticka A, Honzikova N, Fiser B, and Novakova S. "Four signal processing techniques for continuous baroreflex determination". *Proceedings of the 22nd Annual EMBS International Conference*, July 23-28, 2000
Lang *et al.* 2005	R.M. Lang *et al.*, "Recommendation for Chamber Quantification: A Report from the American Society of Echocardiography's Guidelines and Standards Committee and the Chamber Quantification Writing Group, developed in conjunction with the European Association of Echocardiography, a branch of the European Society of Cardiology", *J Am Soc Echocardiogr*, 18, pp. 1440-1463, 2005.
Ledley and Lusted 1959	Ledley, R. S. and Lusted, L. B., "Reasoning foundations of medical diagnosis; symbolic logic, probability, and value theory aid our understanding of how physicians reason", *Science*, 130(3366), pp. 9–21, 1959
Lodwick 1965	Lodwick, G. S., "A probabilistic approach to the diagnosis of bone tumors", *Radiol Clin North Am*, 3(3), pp. 487–497, 1965

Macfarlane and Lawrie 1989	Macfarlane PW, and Lawrie TDV. *Comprehensive Electrocardiology: Theory and Practice in Health and Disease*, 1st ed., Vol. 1, 2, and 3. Pergamon Press, New York, 1989
Mainz and Viergever 1998	J.B. Mainz and M.A. Viergever, "A survey on medical image registration", *Med. Image Anal*. 2, no. 3, 1-36, 1998
Makela *et al.* 2001	T. Makela, P. Clarysse, O. Sipila, N. Pauna, Q.C. Pham, T. Katila, and I.E. Magnin, "A review of cardiac image registration methods", *IEEE Trans. Med. Imag*. 21, no. 9, 1011-1021, 2001
Malmivuo 2007a	Malmivuo J. "12-Lead ECG System". http://butler.cc.tut.fi/~malmivuo/bem/bembook/15/15.htm Retrieved February 2007
Malmivuo 2007b	Malmivuo J. "Vectorcardiographic Lead Systems". http://butler.cc.tut.fi/~malmivuo/bem/bembook/16/16.htm. Retrieved February 2007
Marakas 1999	Marakas, G. M., *Decision support systems in the twenty-first century*, Prentice Hall, 1999
McNitt-Gray *et al.* 1995	M.F. McNitt-Gray, H.K. Huang, and J.W. Sayre. "Feature selection in the pattern classification problem of digital chest radiograph segmentation". *IEEE Trans Med Imaging*, 14(3), 537-547, 1995
Mignotte and Meunier 2001	M. Mignotte and J. Meunier, "A multiscale optimization approach for the dynamic contour-based boundary detection issue", *Comput. Med. Imag. Graph.*, 25, pp. 265-275, 2001.
Miller *et al.* 2000	Miller, P. L., Frawley, S. J. and Sayward, F. G. "Informatics issues in the national dissemination of a computer-based clinical guideline: a case study in childhood immunization", *Proc AMIA Symp*, pp. 580–584, 2000
Mitchell *et al.* 2002	S. C. Mitchell, J. G. Bosch, B.P. F. Lelieveldt, R. J. van der Geest, J. H. C. Reiber, and M. Sonka, "3-D Active Appearance Models: Segmentation of Cardiac MR and Ultrasound Images", *IEEE Transactions on Medical Imaging*, 21(9), pp. 1167-1178, 2002
Minsky, 1975	Minsky M.. "A framework for representing knowledge". In P.H. Winston, editor, *The Psychology of Computer Vision*. New York: McGraw-Hill Book Company, 1975
Morganroth 1985	Morganroth J. "Ambulatory Holter electrocardiography: choice of technologies and clinical uses". *Annals of internal medicine* (*Ann. intern. med.*) , vol. 102, no1, pp. 73-81, 1985
Moroni *et al.* 2006	Moroni D., Perner P., Salvetti O., "A general approach to shape characterization for biomedical problems", *Industrial Conference on Data Mining ICDM. Workshop on Mass-Data Analysis of Images and Signals, MDA 2006*, pp. 56-65, 2006
Motik *et al.* 2004	B. Motik, U. Sattler, R. Studer, " Query Answering for OWL-DL with Rules". *Proc. of the 3rd International Semantic Web Conference (ISWC 2004)*, pp. 549-563, 2004
Muñoz *et al.* 2003	X. Muñoz, J. Freixenet, X. Cufí, J. Martí, "Strategies for image segmentation combining region and boundary information", *Pattern Recognition Letters*, 24, pp. 375-392, 2003.
Niemann *et al.*, 1990	Niemann H., Sagerer G.F., Schroder S., Kummert F. "Ernest: A semantic network system for pattern understanding". *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12:9:883-905, 1990
Noble and Boukerroui 2006	J.A. Noble, D. Boukerroui, "Ultrasound image segmentation: a survey", *IEEE Trans. Med. Imag.*, 25, pp. 987-1010, 2006

Novakova *et al.* 2000	Novakova Z, Honzikova N, Hrstkova H, Fiser B, and Vaclavkova P. "Baroreflex sensitivity and a 24-hour blood pressure profiles in children and adolescents with essential hypertension". *Scripta Medica* (BRNO) – 73 (3):201-208, September 2000
NYMC 2007	New York Medical College. "Circulatory Autonomic Testing. Autonomic Testing, Heart Rate variability, Blood Pressure Variability, and the Baroreflex". http://www.nymc.edu/fhp/centers/syncope/Heart%20rate%20variability,%20blood%20pressure%20variability%20and%20the%20Baroreflex.htm.
O'Connor 2007	M. O'Connor, " Efficiently Querying Relational Databases using OWL and SWRL", http://ontolog.cim3.net/file/work/DatabaseAndOntology/2007-02-08_MartinOConnor/, 2007
Ohno-Machado *et al.* 1998	Ohno-Machado, L., Gennari, J. H., Murphy, S. N., Jain, N. L., Tu, S. W., Oliver, D. E., *et al.*, "Guideline interchange format: a model for representing guidelines", *J Am Med Inform Assoc*, 5(4), pp. 357–372, 1998
Osher and Paragios 2003	S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision and Graphics*, Springer Verlag, 2003
Osheroff *et al.* 2004	Osheroff J.A., Pifer E.A., Sittig D.F., Jenders R.A., Teich J.M. *Clinical decision support implementers' workbook*. Chicago: HIMSS, 2004
Otsu 1979	N. Otsu, "A threshold selection method from gray-level histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, 9, pp. 62-66, 1979.
Paragios 2002	N. Paragios, "A Variational Approach for the Segmentation of the Left Ventricle in Cardiac Image Analysis", *Int. J. of Comp. Vision*, 50(3), pp. 345-362, 2002.
Paragios and Deriche 1999	N. Paragios, R. Deriche, "Unifying boundary and region-based information for geodesic active tracking". In: *International Conference on Computer Vision and Pattern Recognition*, Colorado, vol. II, pp: 422-427, 1999
Peleg, *et al.* 2000	Peleg, M., Boxwala, A. A., Ogunyemi, O., Zeng, Q., Tu, S., Lacson, R., *et al.*, "GLIF3: the evolution of a guideline representation format", *Proc AMIA Symp*, pp. 645–649, 2000
Peleg *et al*, 2002	Peleg M., Yeh I., Altman R.B. "Modelling biological processes using workflow and Petri Net models" Bioinformatics Vol. 18 no. 6, pp. 825-837, 2002
Penicka *et al.* 2004	Penicka M, Bartuneck J, De Bruyne B, Vanderheyden M, Goethals M, De Zutter M, *et al*. "Improvement of left ventricular function after cardiac resynchronization therapy is predicted by tissue Doppler imaging echocardiography". *Circulation*, vol. 109, 978-83, 2004
Pennec *et al.* 2000	X. Pennec, N. Ayache, and J.P. Thirion, *Handbook of medical imaging:processing and analysis*, ch. Landmark-based registration using features identified through differential geometry, pp. 499-513, Academic press, 2000
Pipberger, Stallmann *et al*. 1963	Pipberger, H. V., Stallmann, F. W., Yano, K. and Draper, H. W., "Digital computer analysis of the normal and abnormal electrocardiogram", *Prog Cardiovasc Dis*, 5, pp. 378–392, 1963
Power 1997	Power, D. J.,"What is a DSS?", *The On-Line Executive Journal for Data-Intensive Decision Support*, 1(3), 1997.
Power 2000	Power, D. J., "Web-based and model-driven decision support systems: concepts and issues", *Proceedings of the Americas Conference on Information Systems*, Long Beach, California, 2000
Power 2002	Power, D. J., *Decision support systems: concepts and resources for managers*, Quorum Books, 2002
Reich and Kapeliuk 2005	Y. Reich, A. Kapeliuk, *Decision Support Systems*, 41(1), pp 1-19, 2005

Reinolds *et al.* 2004	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.
Resnic *et al*. 2000	Resnic, F. S., Popma, J. J. and Ohno-Machado, L., "Development and evaluation of models to predict death and myocardial infarction following coronary angioplasty and stenting". *Proc AMIA Symp*, pp. 690–693, 2000
Sakkalis *et al.* 2003	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*, 30, 2003
Sauter 1997	V. L. Sauter, *Decision support systems: an applied managerial approach*, Wiley, 1997
Sermesant *et al.* 2003	M. Sermesant, C. Forest, X. Pennec, and N. Ayache, "Deformable biomechanical models: Application to 4D cardiac image analysis", *Medical Image Analysis*, 7, pp. 475-488, 2003
Sethian 1996	J. Sethian, *Level set methods: Evolving interfaces in geometry, fuid mechanics, computer vision and material science*, Cambridge University Press, 1996
Sezgin and Sankur 2004	M. Sezgin, B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation", *Journal of Electronic Imaging*, 13, pp. 146-165, 2004
Shiffman *et al*. 1999	Shiffman, R. N., Liaw, Y., Brandt, C. A. and Corb, G. J., "Computer-based guideline implementation systems: a systematic review of functionality and effectiveness", *J Am Med Inform Assoc*, 6(2), pp. 104–114, 1999
Shortliffe *et al*., 1975	Shortliffe, E. H., Davis, R., Axline, S. G., Buchanan, B. G., Green, C. C. and Cohen, S. N., "Computer-based consultations in clinical therapeutics: explanation and rule acquisition apabilities of the MYCIN system" *Comput Biomed Res*, 8(4), pp.303–320, 1975
Sowa, 2000	Sowa J.F. "Knowledge Representation: Logical, Philosophical, and Computational Foundations". Brooks/Cole: New York, 2000
Silver 1991	M. Silver, *Systems that support decision makers: description and analysis*, Wiley, 1991
Sprague and Carlson 1982	R.H. Sprague and E. D. Carlson, *Building effective decision support systems*, Prentice-Hall, 1982
Sprague and Watson 1993	R.H. Sprague and H. J. Watson, *Decision support systems: putting theory into practice*, Prentice Hall, 1993
Stadilis *et al.* 1999	G. Stalidis, N. Maglaveras, A. Dimitriadis, and C. Pappas, "Using Learning Classification and Four-Dimensional Parametric Modeling for the Analysis of Myocardial Thickening", *Computers in Cardiology*, 26, 1999, pp. 659-662.
Stallman 1994	R. Stalmann "Why Software Should Not Have Owners", http://eu.conecta.it/paper/Bibliography.html#stallman:why-software-not-owners:98, 1994
Stanhope 2002	P. Stanhope, *Get in the Groove: building tools and peer-to-peer solutions with the Groove platform*, Hungry Minds Pub., New York, 2002
Swedberg *et al.* 2005	K. Swedberg et al., The Task Force for the diagnosis and treatment of CHF of the European Society of Cardiology, "Guidelines for the diagnosis and treatment of Chronic Heart Failure: full text (update 2005)", *European Heart Journal*, 45 pages, 2005.
Talmon 1983	Talmon JL. *Pattern recognition of the ECG, a structured analysis.* PhD Thesis Free University of. Amsterdam, Utrecht 1983
Ten Haken *et al*. 1995	Ten Haken, R. K., Fraass, B. A., Kessler, M. L. and McShan, D. L., "Aspects of enhanced three-dimensional radiotherapy treatment planning", *Bull Cancer*, 82(Suppl 5), pp. 592s–600s, 1995

Thirion 1995	J-P. Thirion, *Fast non rigid matching of 3D medical images*, Tech. Report 2547, INRIA, 1995
Tikkanen 1999	Tikkanen P. *Characterization and application of analysis methods for ECG and time interval variability data*. Academic Dissertation at the Faculty of Science, University of Oulu, 1999
Tjeerdsma 2001	Tjeerdsma G. *Value of neurohormonal and autonomic parameters for the assessment of the severity and prognosis in chronic heart failure*. Dissertation at the University of Groningen, 2001
Trier and Jain 1995	O.D. Trier, A.K. Jain, "Goal-directed evaluation of binarization methods", *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 17, pp. 1191-1201, 1995.
Turban 1995	E. Turban, *Decision support and expert systems: management support systems*, Prentice Hall, 1995
Turban 2005	E.Turban, J.E. Aronson, and T.P. Liang, (2005), *Decision Support Systems and Intelligent Systems*, Pearson Education, NJ, 2005
Walker *et al.*, 1990	Walker A., McCord M., Sowa J.F., Wilson W.G.. "Knowledge Systems and Prolog", Second Edition, Addison-Wesley, 1990
Warner, Toronto *et al*. 1964	Warner, H. R., Toronto, A. F. and Veasy, L. G., "Experience With Baye's Theorem For Computer Diagnosis Of Congenital Heart Disease". *Ann N Y Acad Sci*, 115, pp. 558–567, 1964
Willems *et al.* 1990	Willems JL, et al. "Common standards for quantitative electrocardiography: goals ad main results. CSE Working Party", *Methods Inf Med*. 29(4), pp 263-71, 1999
Yan *et al.* 2005	F. Yan, H. Zhang, C.R. Kube, "A multistage adaptive thresholding method", *Pattern Recognition Letters*, 26, pp. 1183-1191, 2005
Woods, 1975	Woods W.A. "What's in a link: foundations for semantic networks". In D.G. Bobrow and A.M. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 35-82. New York: Academic Press, 1975
Yu *et al.* 2004	Yu CM, Fung WH, Zhang Q, Chan CK, Chan YS, Lin H, et al. "Tissue Doppler imaging is superior to strain rate imaging and postsystolic shortening on the prediction of reverse remodeling in both ischemic and nonischemic heart failure after cardiac resynchronization therapy". *Circulation, vol.*110, 66-73, 2004
Yue *et al.* 1995	Z. Yue, A. Goshtasby, and W. Ackerman. "Automatic detection of ribs borders in chest radiographs". *IEEE Trans Med Imaging*, 14(3), 525-536, 1995
Zareba *et al.* 1998	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*, 3(1), no3, pp. 261-262, 1998
Zohgbi *et al.* 2003	Zohgbi *et al.* American Society of Echocardiography's Nomenclature and Standards Committee and The Task Force on Valvular Regurgitation. "Recommendations for Evaluation of the Severity of Native Valvular Regurgitation with Two-dimensional and Doppler Echocardiography", *Journal of the American Society of Echocardiography*, 777-802, 2003
Zywietz 2007	Zywietz Chr. "A Brief History of Electrocardiography – Progress through Technology". http://www.openecg.net/Tutorials/A_Brief_History_of_Electrocardiography.pdf
Zywietz and Fischer 2004	Zywietz Chr, and Fischer R. "Integrated Content and Format Checking for Processing of SCP ECG Records", *Proceedings of Computers in Cardiology*, 31, pp. 37-40, 2004
Zywietz *et al.* 2001	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* 28, pp. 189-192, 2001

Miscellaneous Web Resources

Advocate Health	Advocate Health Care – Cardiopulmonary Stress Test. http://www.advocatehealth.com/system/services/heart/testing.html#cpstress
AMIA	http://www.amia.org/inside/initiatives/cds/cdsroadmap.pdf
CENORM	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
D2R MAP	D2R MAP - Database to RDF Mapping Language and Processor http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2rmap/D2Rmap.htm
DartGrid	DartGrid data integration toolkit http://ccnt.zju.edu.cn/projects/dartgrid/
Data Formats	Scientific data formats. http://www.dpmi.tugraz.at/~schloegl/matlab/eeg/#EDF
DICOM	Digital Communication in Medicine (DICOM) Homepage. http://medical.nema.org/
DIG	DIG description logic reasoner interface http://sourceforge.net/projects/dig
Eercise ECG Testing	Exercise ECG testing– Patient UK. http://www.patient.co.uk/showdoc/40000517/
FINAPRES	FMS Finapres Medical Systems – the company for non-invasive blood pressure measurements systems. http://www.finapres.com/
Hypertext	http://en.wikipedia.org/wiki/Hypertext
IEEE1073	IEEE 1073: Medical Device Communication. http://www.ieee1073.org/standards/1073standards.html. (last visited February 2007)
MedLine	MedLine Plus Medical Encyclopedia: Holter monitor (24h). http://www.nlm.nih.gov/medlineplus/ency/article/003877.htm
MIT_BIH	MIT_BIH Arrhythmia Database Directory. http://www.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm
Open Source	http://en.wikipedia.org/wiki/Open_source
Open Source Organization	http://www.opensource.org/licenses/index.php
Openclinical	http://www.openclinical.org
OpenECG	The OpenECG portal. http://www.openecg.net. 1)How to Implement SCP – Part I. http://www.openecg.net/tutorial1/index.html. 2)How to Implement SCP – Part II. http://www.openecg.net/tutorial2/index.html
OWL	Web Ontology Language (OWL) - W3C http://www.w3.org/2004/OWL/
PhysioBank	An Introduction to the PhysioBank Archives. http://www.physionet.org/physiobank/physiobank-intro.shtml#records
PhysioNet	PhysioNet: Research Resources for Complex Physiologic Signals. http://www.physionet.org/
RDF	Vocabulary Description Language 1.0: RDF Schema - Dan Brickley (W3C), R.V. Guha (IBM) http://www.w3.org/TR/rdf-schema/

RDF/XML	Syntax Specification (Revised) - Dave Beckett (University ofBristol) http://www.w3.org/TR/rdf-syntax-grammar/
RDF-Italian Translation	(RDF) Italian translation – Andrea Marchetti (IIT-CNR), Massimo Martinelli (ISTI-CNR), Enrico Seta (Camera dei Deputati) http://www.xml.it:23456/RDF/REC-rdf-syntax-19990222-it.html
Rule Interchange Format	Rule Interchange Format - W3C Working Group http://www.w3.org/2005/rules/Overview.html
Semantic Web	http://www.w3.org/2001/sw/
SPARQL	Query Language for RDF - Eric Prud'hommeaux (W3C), Andy Seaborne (Hewlett-Packard Laboratories, Bristol) http://www.w3.org/TR/rdf-sparql-query/
SWRL	SWRL: A Semantic Web Rule Language Combining OWL and RuleML – Ian Horrocks (Network Inference), Peter F. Patel-Schneider (Bell Labs Research, Lucent Technologies), Harold Boley (National Research Council of Canada), Said Tabet (Macgregor, Inc.), Benjamin Grosof (Sloan School of Management, MIT), Mike Dean (BBN Technologies) http://www.w3.org/Submission/SWRL/
UNICODE	What is Unicode http://www.unicode.org/standard/WhatIsUnicode.html
URI	http://www.ietf.org/rfc/rfc1630.txt
XML	(XML) 1.0 (Fourth Edition) - Tim Bray (Textuality and Netscape), Jean Paoli(Microsoft), C. M. Sperberg-McQueen (W3C), Eve Maler (Sun Microsystems, Inc.), François Yergeau http://www.w3.org/TR/xml/

List of Abbreviations

6MWT	6Minute Walking Test
AAM	Active Appearance Model
ACC	American College of Cardiology
ACR	American College of Radiology
AHA	American Heart Association
AI	Artificial Intelligence
AMIA	American Medical Informatics Association
ANN	Artificial Neural Network
AR	Aortic Regurgitation
ARMA	AutoRegressive/Moving Average
AS	Aortic Stenosis
ASE	American Society of Echocardiography
ATS	American Thoracic Society
AV	Aortic Valve
BIA	BioImpedance
BPV	Blood Pressure Variability
BRS	Baroreflex Sensitivity
CAD	Computer Aided Design
CAM	Cardiac Autonomic Modulation
CDSS	Clinical Decision Support System
CNR	Italian National Research Council
CPX	Cardiopulmonary Stress Testing
CRT	Cardiac Resynchronization Therapy
CT	Computed Tomography
CWD	Continuous Wave Doppler
DB	Data Base
DBMS	Data Base Management System
DFT	Discrete Fourier Transform
DICOM	Digital Imaging and Communications in Medicine
DSS	Decision Support System
ECG	ElectroCardioGram
ED	End diastole
EF	Ejection Fraction
EIS	Executive Intelligent Systems
ES	End systole
ESC	European Society of Cardiology
FDA	US Food and Drug Administration
FFD	Free Form Deformation
FFT	Fast Fourier Transform
GDSS	Group DSS
GFB	Geometric Feature Based
HF	High Frequency
HF	Heart Failure
HFP	HEARTFAID platform
HR	Heart Rate

HRV	Heart Rate Variability
ICA	Independent Component Analysis
IDSS	Intelligent DSS
IPT	Image Processing Toolbox
IVC	Inferior Vena Cava
KDD	Knowledge Discovery in Databases
LA	Left Atrium
LF	Low Frequency
LOINC	Logical Observation Identifiers, Names, and Codes
LV	Left Ventricle
LVOT	Left Ventricle Outflow Tract
MBMS	Model Base Management System
MRI	Magnetic Resonance Imaging
MV	Mitral Valve
NASPE	North American Society of Pacing and Electrophysiology
NEMA	National Electrical Manufacturer's Association
NYHA	New York Heart Association
NYMC	New York Medical College
ODSS	Organizational DSS
PCA	Principal Component Analysis
PET	Positron Emission Tomography
PWD	Pulsed Wave Doppler
RDF	Resource Description Framework
SIB	Standard Intensity Based
SPT	Spirometry
STFT	Short Time Fourier Transform
STX	Exercise Test
SVM	Support Vector Machine
SWRL	Semantic Web Rule Language
TAPSE	Tricuspid Annular Plane Systolic Excursion
TDI	Tissue Doppler Imaging
TEE	TransEsophageal Echocardiography
TTE	TransThoracic Echocardiography
TV	Tricuspidal Valve
UMLS	Unified Medical Language System
URI	Universal Resource Identifiers in WWW
US	UltraSound
VLF	Very Low Frequency
WT	Wavelet Transform
XML	eXtensible Markup Language