HELSINKI UNIVERSITY OF TECHNOLOGY

Department of Computer Science
Laboratory of Computer and Information Science

Kari Sikiö

Design and Implementation of Biosignal Monitoring and Analysis Software

This thesis has been submitted for official examination for the degree of Master of Science in Computer Science.

Supervisor of the thesis: Prof. Olli Simula
Instructor of the thesis: M.Sc. Sakari Lukkarinen

Espoo 27.5.1999

HELSINKI UNIVERSITY OF TECHNOLOGY **ABSTRACT OF THE MASTER'S THESIS**

Author: Kari Sikiö

Name of the thesis: Design and Implementation of Biosignal Monitoring and Analysis Software

Date: 27.5.1999 **Number of pages:** 75

Department: Department of Computer Science and Engineering
Professorship: Tik-115, Information Science

Supervisor: Prof. Olli Simula
Instructor: M.Sc. Sakari Lukkarinen

Motion of the human heart gives rise to vibrations and sounds that can be acquired from the surface of the chest. These sounds contain information regarding the condition of the heart. There are two principal methods to investigate heart sounds: cardiac auscultation and phonocardiography. In auscultation, a stethoscope is used for examination; in phonocardiography, some kind of device is used for recording the sounds. Cardiac auscultation is widely practiced today as the primary testing method for finding out what other methods of cardiac examination are needed, if any. Phonocardiography has become more or less obsolete, and is only rarely used.
There are many reasons for this; one is the rapid development of other methods for cardiac examination. Another reason is that there hasn't been powerful and easy-to-use phonocardiography systems available.

This master's thesis describes the development of Amore, a heart sound monitoring and analysis program, which together with PhonoScope phonocardiography pre-amplifier [1], forms a complete phonocardiography system. Amore has been written for 32-bit Microsoft Windows operating systems and can be used for monitoring, recording and analyzing practically any audio signals. Conversions from analog to digital domain and vice versa are performed with any soundcard complying with the Windows multimedia interface. Sound documents are stored in the computer's harddisk using a standardized file format

Software-aided heart sound analysis has several advantages compared to traditional cardiac auscultation using a stethoscope. Maybe the most obvious one is the improved reliability of diagnoses: for example, murmurs at very low frequencies may not be heard, but can be clearly seen in a spectrogram representation of the sound. Another significant advantage is that heart sounds can be saved as a part of the documentation, such as patient information and diagnosis, in a single file. When necessary, this file can easily be sent as an e-mail attachment to a cardiology specialist, who can listen to the heart sound, examine the graphs, compare data to examples in a local phonocardiography database and diagnose the finding. Amore and PhonoScope, or even the software alone, provides a powerful tool for the educational field as well.

Keywords: Audio, Auscultation, Biosignal, Cardiography, Heart sound, Phonocardiography, PhonoScope

TEKNILLINEN KORKEAKOULU **DIPLOMITYÖN TIIVISTELMÄ**

Tekijä:	Kari Sikiö	
Työn nimi:	Biosignaalien monitorointi- ja analysointiohjelman suunnittelu ja toteutus	
Päivämäärä:	27.5.1999	**Sivumäärä:** 75
Osasto:	Tietotekniikan osasto	
Professuuri:	Tik-115, Informaatiotekniikka	
Työn valvoja:	Prof. Olli Simula	
Työn ohjaaja:	DI Sakari Lukkarinen	

Sydämen sykintä aiheuttaa mitattavissa olevaa rintakehän värärähtelyä ja ääniä, jotka sisältävät tietoa sydämen kunnosta. Sydänäänien tutkimiseen käytetään kahta perusmenetelmää: auskultointi ja fonokardiografia. Auskultoinnissa tutkimusvälineenä on stetoskooppi, kun taas fonokardiografiassa äänet tallennetaan jollakin laitteella. Nykyään auskultointia käytetään yleisesti ensisijaisena tutkimusmenetelmänä pyrittäessä selvittämään onko sydän terve ja millaisia jatkotutkimuksia tehdään, jos niihin on tarvetta. Fonokardiografiaa käytetään jokseenkin harvoin. Tähän on monia syitä; yksi niistä on muiden tutkimusmenetelmien nopea kehitys. Toinen syy on ollut tehokkaiden ja helppokäyttöisten fonokardiografiajärjestelmien puuttuminen.

Tämä diplomityö käsittelee Amore-nimisen sydänäänten monitorointi ja analysointi-ohjelman kehitystä. Yhdessä PhonoScope -fonokardiografiavahvistimen [1] kanssa se muodostaa kokonaisen fonokardiografiajärjestelmän. Amore on kirjoitettu 32-bittisille Microsoft Windows käyttöjärjestelmille. Sitä voi käyttää minkä tahansa äänisignaaleiden monitorointiin, äänitykseen ja analysointiin. Muunnokset analogisesta digitaaliseen ja päinvastoin tehdään jollain Windows:n multimedia-rajapintaan soveltuvalla äänikortilla. Äänidokumentit tallennetaan tietokoneen kovalevylle standardisoidussa formaatissa.

Tietokoneavusteinen sydänäänianalyysi tarjoaa useita etuja perinteiseen stetoskooppi-auskultointiin verrattuna. Ehkä ilmeisin niistä on diagnoosien suurempi luotettavuus: esimerkiksi erittäin matalalla taajuusalueella esiintyvät sivuäänet saattavat jäädä kuulematta, mutta ne voidaan helposti nähdä äänen spektrogrammiesityksessä.
Toinen merkittävä etu on mahdollisuus tallettaa äänet yhdessä potilastietojen ja diagnoosin kanssa yhdeksi dokumenttitiedostoksi. Tarpeen tullen tällainen tiedosto on helppo lähettää vaikkapa sähköpostin liitteenä konsultoivalle kardiologille, joka voi kuunnella sydänääniä, tutkia kuvaajia ja diagnosoida löydöksen käyttäen apunaan keräämiänsä esimerkkitapauksia. Amore, Phonoscope:n kanssa tai ilman sitä, on varteenotettava työkalu myös opetuskäyttöön.

Avainsanat: audio, auskultointi, biosignaali, fonokardiografia, kardiografia, Phonoscope, sydänääni

Acknowledgments

This master's thesis describes the results of four and a half years of work for developing a software to be used as a part of a phonocardiography system..

I would like to thank Sakari Lukkarinen at Medivisio Ltd. for offering me this challenge, for trusting and encouraging me along the project, and for making the work economically possible. I am grateful to Sakari for his patience and understanding, and also for acting as my supervisor for the thesis.

Special thanks go to cardiologists Anna-Leena Noponen and Milad El-Segaier for testing the software in their work and giving me lots of valuable feedback, and to my brother and sister for their unforgettable support and care.
Very special thanks belong to Katariina Valtonen for forwarding the announcement of "multimedia documentation job" to me in autumn 1994

Finally, I would like to give thanks to my parents for supporting me in so many ways, to Johanna just for being there, and to everybody who has helped me to finish this work or otherwise deserves my gratitude.

Helsinki, May 1999

Kari Sikiö

Praise the Lord!

For it is good to sing praises to our God;

for he is gracious, and a song of praise is seemly.

The Lord builds up Jerusalem;

he gathers the outcasts of Israel.

He heals the brokenhearted,

and binds up their wounds.

He determines the number of the stars,

he gives to all of them their names.

Great is our Lord, and abundant in power;

his understanding is beyond measure.

Psalm 147:1-5

Table of contents

Glossary

A/D	Analog to Digital (conversion)
auscultation	the act of listening, either directly or through a stethoscope or other instrument, body sounds as a method of diagnosis
amplitude graph	signal amplitude vs. time presentation
anesthetic gases	easily inflammable gases used in anesthesia
cardiology	the study of the heart and its functions in health and disease
CD-ROM	Compact Disk - Read Only Memory
D/A	Digital to Analog (conversion)
ECG	electrocardiogram; detecting and recording the minute differences in potential caused by heart action
FFT	Fast Fourier Transform
FIR	Finite Impulse Response; type of digital filter
geriatrics	the medical science dealing with the diseases, debilities, and care of aged persons
IIR	Infinite Impulse Response; type of digital filter
MPEG	abbreviation for Motion Picture Expert Group compression standards
out-patient	a patient receiving treatment at a hospital, but not being an inmate
PC	Personal Computer
PCG	phonocardiography; recording of heart sounds
pediatrics	science dealing with the medical and hygienic care of children or with the diseases of children
phonocardiograph	an instrument for graphically recording heart beat sounds
policlinic	a hospital department at which out-patients are treated
RIFF	Resource Interchange File Format for multimedia files by Microsoft and IBM

s/n-ratio	Signal-to-Noise ratio, describes the amount of unwanted noise in a signal
soundcard	an add-on device inside a computer, which produces and receives analog sound signals
spectrogram	time-frequency-intensity presentation of a signal
teleconsultation	remote consultation utilizing telecommunication
THD	Total Harmonic Distortion, describes the amount of distortion in a signal
transducer	a device that receives energy from one system and retransmits it, often in a different form, to another
WAV	WAVe file format (a sub-type of RIFF), used by Windows for audio storage
Windows	operating system concept developed by Microsoft; Windows'95, Windows'98 and Windows NT are separate operating systems sharing the same basic concept

1. Introduction

1.1 Background

In the beginning, there was just the idea of multimedia documentation application: heart beat sounds would be saved along with other documentation, such as patient information and diagnosis. From the programmer's point of view, the most essential problem was to develop an application with sufficiently attractive and convincing user interface to be accepted by cardiologists.

First attempt towards this goal was HeartBeat, a 16-bit application for Microsoft Windows. The basic idea was to provide a single window, resembling a document page, for both patient information and an audio graph for one listening point at a time. Recordings at several listening points could be included in a single document.

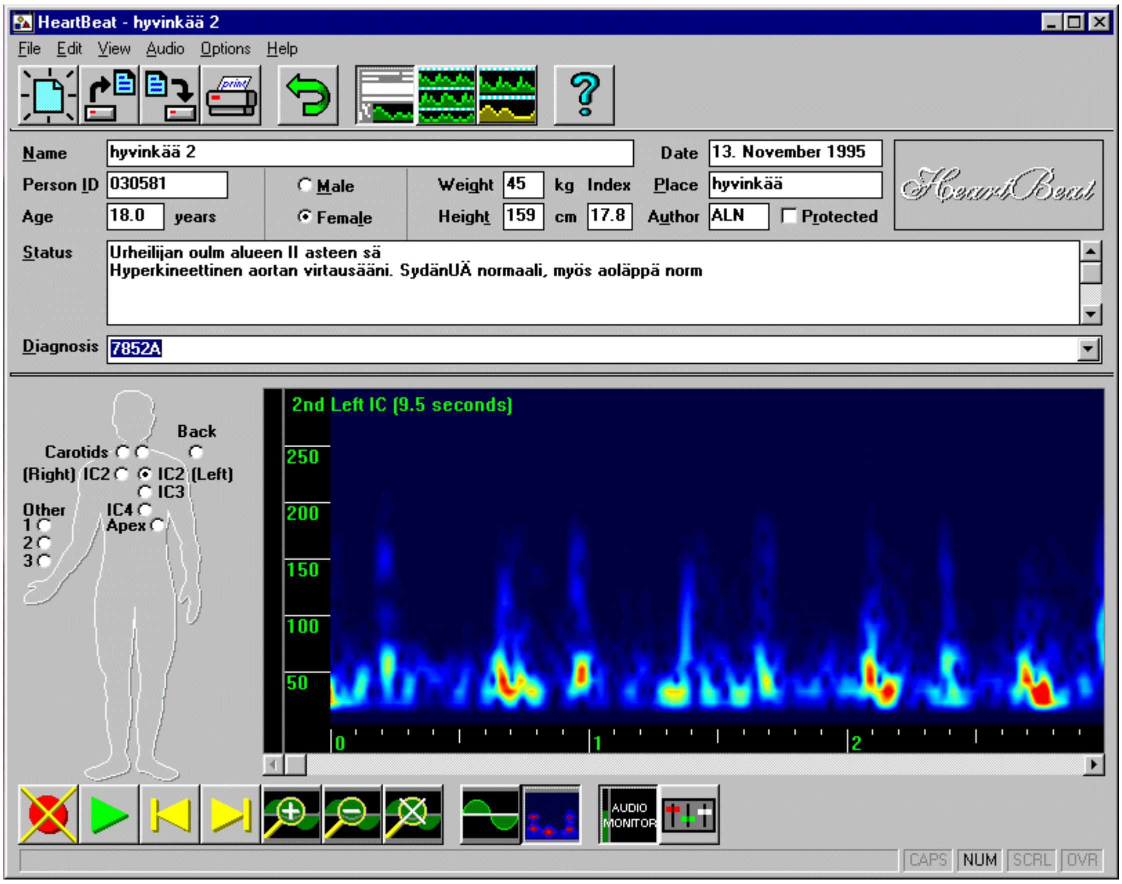


***Figure 1.1**: HeartBeat v.3.2, main view with spectrogram of the sound at 2.intercostal*

Later on, a possibility to view multiple graphs simultaneously was added. Both amplitude and spectrogram representations of the sound were available already in early versions. After a year of development HeartBeat's user interface had reached its final form, including support for an external control device that could be connected to the serial communication port of a PC. The main window of HeartBeat is presented in figure 1.1.

The results gained with HeartBeat were promising. However, it turned out that the concept of multiple recordings in a single document, at the cost of more complex and less intuitive user interface, didn't serve the purpose well; typically recording at only one listening point was needed for the diagnosis. A new approach of single listening point documentation was taken, and HeartBeat's destiny became to be the predecessor of Amore, **A**uscultation **Mo**nitor and **Re**corder. The goal was to develop powerful and yet easy-to-use phonocardiography software. The results are presented later in this document.

1.2 Motivation

Forms of innocent murmur is said to be heard at one time or another in 70 to 85% of all children [2]. Although heart murmurs are easily heard, classification of heart murmurs into innocent and pathological by auscultation is strongly dependent on listener's experience and training [3]. Nowadays, ultrasound is thought to be the most reliable method in heart murmur diagnostics. However, ultrasound equipment are mainly found in hospitals, and a professional cardiologist is needed to evaluate the results.

Part of the motivation for this work is based on the fact that the phonocardiographic signal has much more clinically useful information than can be assessed by auscultation alone [4]. This could, properly utilized, significantly improve the possibilities to diagnose innocent murmurs already in the primary health care. Also, today and almost certainly even more in the future, electrically exchangeable documents would provide a time and cost efficient way for cardiologic consultation when that is necessary.

2. Requirements Specification

Software specification is one of the fundamental activities common to all software projects. In the requirements specification, the functionality of the software and constraints on its operation are defined. The accuracy and modifiability level of the specification depends on the selected software development model. In this project, evolutionary development [5] is used: the activities of specification, development and validation are interleaved. The initial, rather abstract specification is refined along the development process in order to produce a system which satisfies the customer's needs. In this chapter, the requirements specification for the currently available product (fig. 2.1) is presented. Although the software is only a part of the system, full specification is included here to help the reader understand how the parts are related to each other.

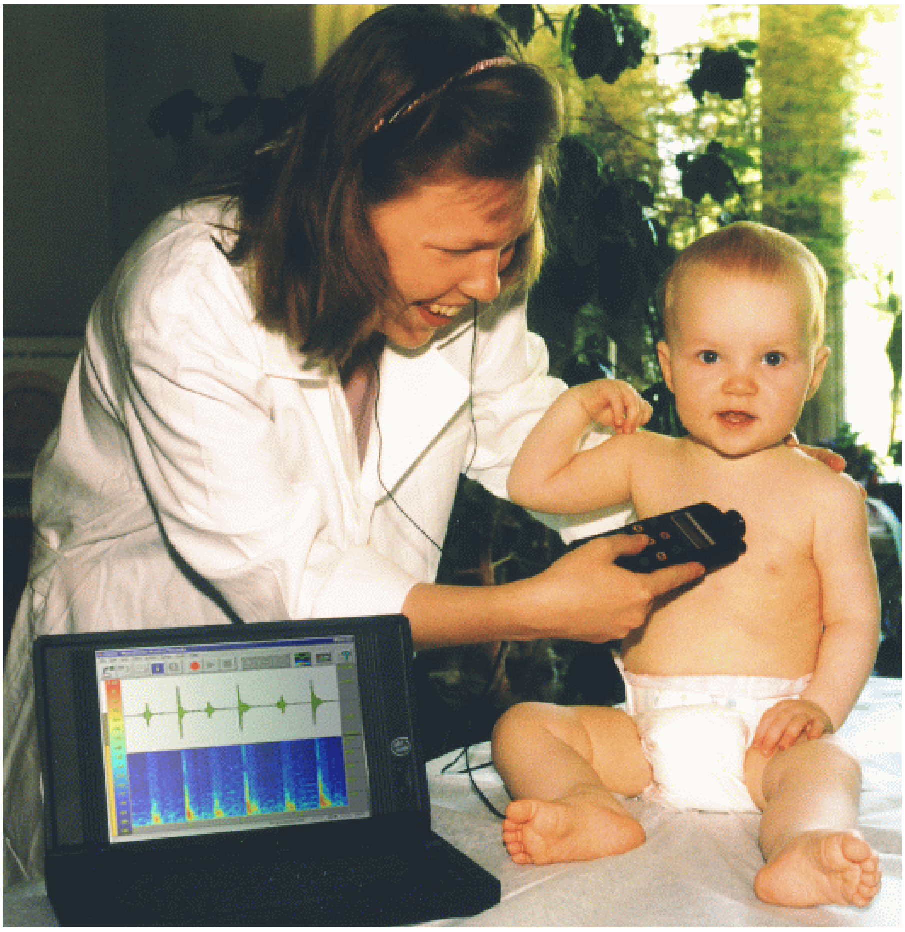


Figure 2.1*: Auscultation with the PhonoScope system*

2.1 General description

PhonoScope - Auscultation Monitoring and Analysis System is an active diagnostic medical equipment designed for listening, recording, documenting and analyzing auscultation findings. The system is used by physicians and medical personnel with special training. The purpose of the system is to provide an objective tool for analysis of auscultation findings.

2.1.1 System Overview

The PhonoScope system consists of phonocardiographic transducer(s) and an amplifier, headphones, a computer equipped with a soundcard, and an auscultation analysis software. The system is based on the computer, which serves as a display for the graphical presentations, as a recorder for the phonocardiographic and other optional signals, as a memory and disk storage for the documentation data. The computer also offers a way to send the auscultation data to other computers and printing devices.

Figure 2.2*: Headphones and a transducer*

A phonocardiographic transducer (fig. 2.2) picks up the sounds within the patient’s body and transforms them into electrical signals. The transducer is connected to an amplifier, which amplifies the signals to a suitable level for a soundcard. The soundcard transforms the signals into digital domain for the analysis software. The software, running under Windows operating system, is intended for monitoring, recording and analyzing phonocardiographic and other, optional signals. Only basic skills of using Windows are needed to utilize all of software’s functionality.

2.1.2 End Users

PhonoScope is intended to be used by medical physicians and specialized nurses. End users can be divided to following groups: general practitioners, pediatricians, pediatric cardiologists, cardiologists, clinical physicians and internal medicine physicians. A separate group of users is formed by the pediatric, cardiological and clinical physician teachers, who teach auscultation to medical candidates. User groups are analyzed further in this sub-chapter.

General physician

- Uses the product weekly for listening and recording the auscultation findings
- Sends the findings of pediatric and geriatric patients to a specialist to get a second opinion
- Knows how to listen, record, document and send the documents to the specialists
- Needs basic user guidance (2 h)
- Benefits from the possibility to get a second opinion from a specialist

Specialist for pediatrics or internal medicine

- Uses the product daily for listening, recording, analyzing the auscultation findings
- Acts as a consultant for general practitioners
- Knows how to listen, record, document, analyze, and send the documents
- Needs basic user guidance (2 h) and special training in graphical analyzing (2 h)
- Benefits from the possibility to hear and see the graph of findings simultaneously, and to select the policlinical patient material more accurately with better basic screening and teleconsultation

Senior specialist for cardiology or pediatric cardiology

- Uses the product daily for examining patients and receiving consultation requests from general practitioners and specialists
- Knows how to listen, record, document, analyze, and send the documents
- Needs basic guidance (2h), special training in analyzing (2h) and teleconsultation (1h)
- Benefits from the possibility to give priority to severe cases based on auscultation findings in teleconsultation requests

Teacher

- Uses the product for teaching auscultation and collecting teaching material
- Knows how to listen, record, document and analyze the findings
- Needs basic guidance (2h), special training in analyzing (2h)
- Interested in recording special cases and using them as teaching material
- Benefits from the possibility to record, replay and visualize the sounds. The students can study the cases without actual patients

2.1.3 Operation environment

In primary health care, the product is intended to be used in out-patient clinics, emergency rooms, health care centers, occupational health care offices, home visits. In special health care, it is to be used in hospital clinics and in bed wards. In university hospitals, it is to be used at the teaching clinics. In the private sector, it is to be used at the pediatric and Cardiological examination rooms. The product can also be used at veterinary clinics.

It is not allowed to use the product in treatment rooms or any other environment where there might be anesthetic gases.

2.2 Functions

2.2.1 Auscultation and monitoring

User wears a pair of headphones and places the transducer against the patients body. During auscultation, the signal can be visually observed on the computer’s display. When needed, the amplification level can be adjusted from the amplifier.

2.2.2 Recording and replaying

Heart sounds can be recorded into the computer’s memory at any time for further analysis and documentation. Recorded sounds can be replayed.

2.2.3 Analyzing

Recorded sounds can be examined and analyzed with the software. Both amplitude and spectrogram graphs are available. The graphs can be viewed in different detail levels. Simple parameters, such as durations, relative amplitudes and frequencies, can be measured from the graphs. Signals can be filtered using digital filters.

2.2.4 Documentation and data management

Auscultation document files can be created, opened and saved. A document file consists of the recorded signals with optional patient and diagnosis information provided by user.

2.2.5 Teleconsultation

Document files can be sent and received using ordinary email communication.

2.2.6 Optional functions

In addition to audio signals, the system is capable of monitoring, recording and analyzing also other analog signals, such as outputs from ECG devices.

2.3 External interfaces

2.3.1 Amplifier and the signal connections

In the front panel of the amplifier, there are following items:

- a gain control for input gain and headphone output level adjustment
- a press button for selecting the function of the gain control
- a press button for selecting the channel(s) that are directed to headphone output
- a five-digit display showing the input gain (in dB) and the headphone output level
- two connectors for line level analog input signals
- a headphone connector

2.3.2 User interface of the software

User interacts with the program via a mouse and a keyboard. Mouse is used for accessing and controlling most of the functionality, such as making selections and measuring properties in the graphs. Keyboard is needed for textual inputs, such as patient's name.

2.3.3 File formats

Document file format complies with the Resource Interchange File Format (RIFF) standard. Signal data are processed in an uncompressed format; no support for compression algorithms, such as MPEG, is provided. WAV is a subformat of the RIFF standard, and is thus supported by the system.

2.4 Other requirements

2.4.1 Hardware requirements

Minimum computer requirements

- Pentium PC, at least 200 MHz processor recommended
- Microsoft Windows'95 or Windows NT
- soundcard with analog input and output, installed as a Windows multimedia device
- 16 MB of RAM (32 MB recommended)
- 640*480 resolution monitor, 256 colors (800*600 with 16-bit colors recommended)
- 3 MB for the application, 100 MB for documents
  (1 GB total hard disk space recommended)
- PhonoScope phonocardiography amplifier or other signal source

Optional hardware requirements

- ECG monitor or plotter with analog output for ECG recordings

2.4.2 Analog input and output signals

Maximum input signal level is 8 Vp-p. Signal frequency band is limited by the soundcard and is typically 20 Hz .. 20 kHz. Also the s/n-ratio and THD typically depend on the quality of the soundcard.

2.4.3 Standards

The system should comply with the following directives and standards for medical equipment:

- EU directive 93/42/ESC
- EN-60601-1-2

2.4.4 Distribution

Every country has a local distributor. Distributors should understand the auscultation procedure. Distributors give the basic and advanced training and respond to inquiries.

2.4.5 Marketing

Academic and clinical studies done with the product will be published in international journals. Satisfied authorities using the product give public statements recommending the product. Infiltration in teleconsultation markets should occur through video conference product providers. Initially the product should be targeted to university clinics for teaching purposes, making future doctors familiar with the product. Later on in the field they long for the advantages of the product and insist it from their employers.

2.4.6 Production

Production is done using suppliers.

2.4.7 Service and warranty

Distributor or qualified service company. Warranty for 1 year.

2.5 Quality plan

The main objectives of the project:

1. An attractive, high-end product
2. High sound quality
3. Intuitive, easy to learn and use
4. Minimal patient examination time
5. Helps to make more objective and accurate diagnoses
6. High mechanical quality
7. Reliable operation of the system
8. Well documented project management

3. Design and Implementation

Amore, **A**uscultation **Mo**nitor and **Re**corder, has been written for 32-bit Microsoft Windows operating systems (Windows'95, '98 and NT). It can be used for monitoring, recording and analyzing any analog signals that are suitable for an A/D conversion performed by a soundcard that complies with the Windows multimedia interface. Signals can be stored on computer's harddisk using a standardized file format.

In most software development projects, design and implementation of the product must go, to some degree, hand in hand. The more detailed design, the better are the odds for it to get modified as the implementation proceeds. This project began as an experimental research for an innovative user interface to provide the functionality described above. As a programmer, it has been a great pleasure for me that the project has preserved its experimental nature all the way: design was kept pretty much open, even the main features could be modified as better ideas came along.

Validation of the correct functioning of Amore has been performed in several stages, primary stage being the real-time testing while generating the code. People who were using the software in their daily work gave not only error reports, but also valuable feedback on the user interface. A thorough discussion of risk analysis and validation procedure would require a separate chapter, and is omitted here.

In this chapter, current design and implementation is presented from four points of view: **system model** as data flow and class hierarchy charts, internal and external **data structures**, **user interface** illustrations, and finally, an overview of **algorithms and functions**. However, the best way to get familiar with the user interface and features of Amore is simply to install it and try it out, utilizing its extensive on-line help features. Just run the Setup.exe file which can be found in the appendix CD-ROM. In the program you can get help by selecting Help|Contents from the menu, or by pressing Shift+F1 and clicking a button (or selecting a command) that you want to know about.

3.1 System Model

There are two major data flow paths in Amore: signal input and signal output. These are illustrated in figure 3.1. Notice the input→output connection of audio through monitoring.

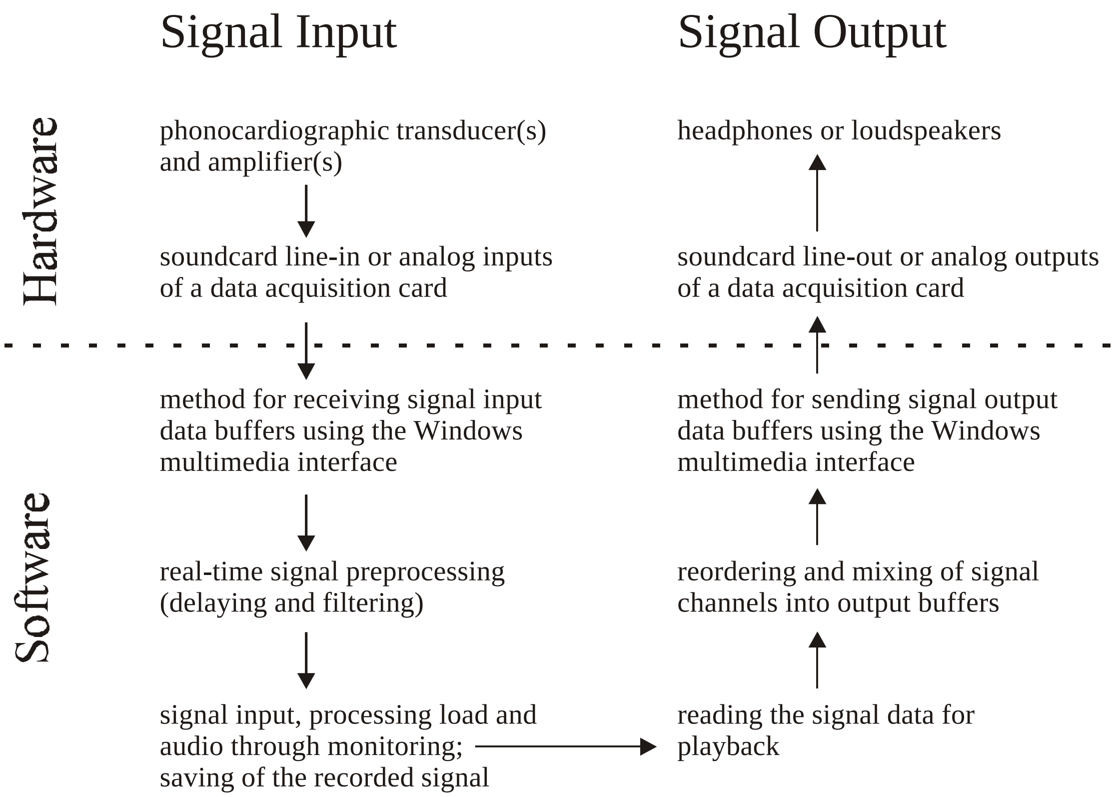


Figure 3.1*: Signal data flow paths*

Amore is programmed in C++ with Borland's BC++ 5.02. The class hierarchy is based on Borland's Object Windows Library (OWL). It determines the roles of inherited classes for Windows applications. Figure 3.2 illustrates this division with class and object relations. Some of the methods and variables of the three main classes (Amore, AmoreFrame and SoundView) are included to give an idea of the functionality provided by corresponding objects. Edges of the OWL base class boxes are rounded. Different lines are used for describing the (class) inheritance and (object) parenthood relations. Some of the minor classes, such as file i/o and small control classes, are not included in this figure.

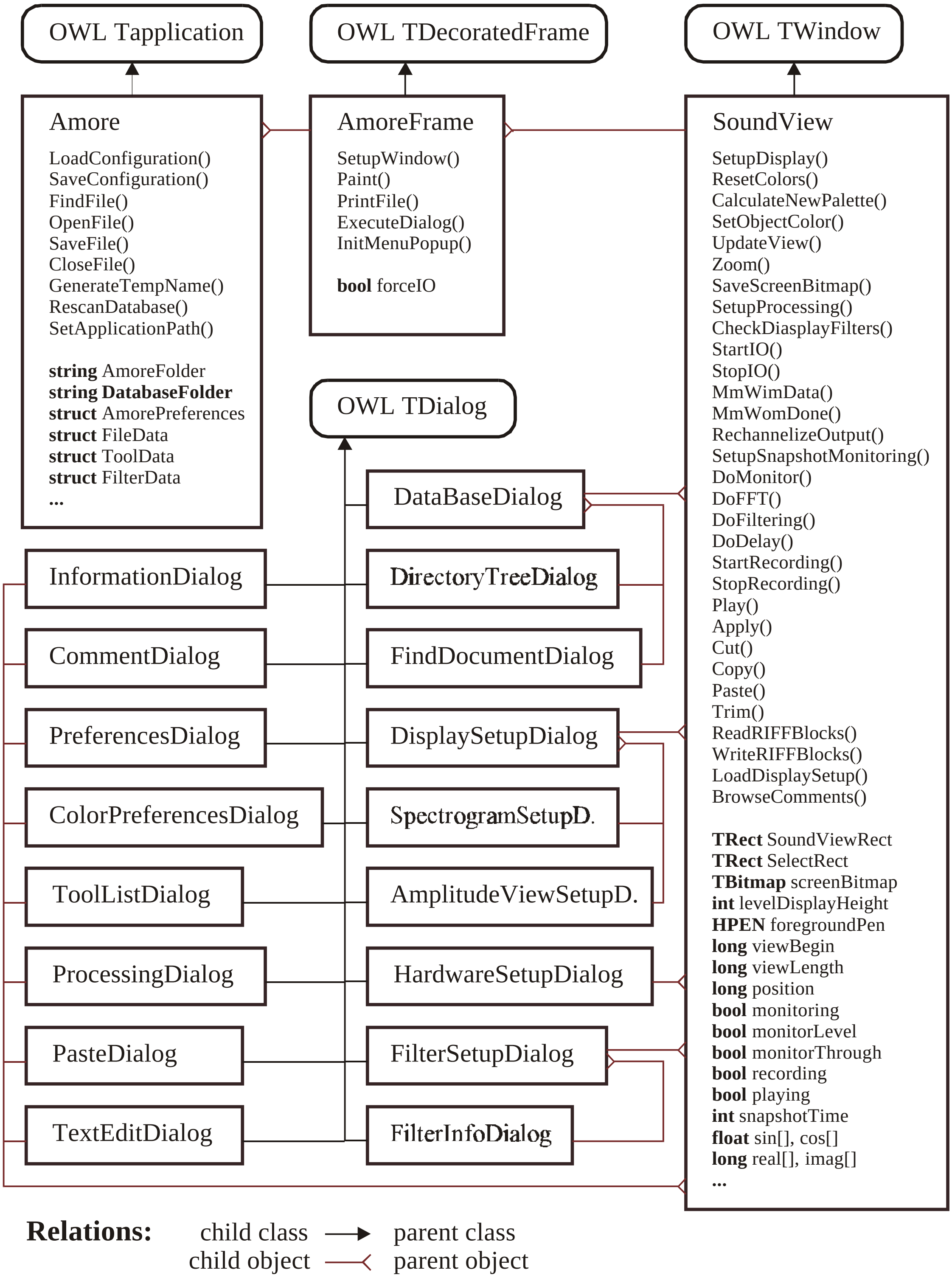


***Figure 3.2**: Class and object hierarchy chart*

3.2 Data Structures

A critical question in software design is the format of data structures: how large structures should be used, how the data are divided in separate structures, and what kind of version control is used for minimizing the problems caused by software evolution. Structure definition solutions can have even more crucial, positively or negatively, effects on the development process than the selected class hierarchy.

In Amore, the data associated with one dialog (such as patient- and diagnosis information, pre- and postprocessing data, filter data, preferences, tools, display setups etc.) are generally stored in a single structure: Version information is needed only for permanently stored data, and is thus included in the file headers instead of the structures themselves.

The data types used in this sub-chapter are explained in table 3.1.

Data type name	Explanation (in C++ terms)
bool	boolean value (**true** or **false**)
char	unsigned character (= **byte**)
word	16-bit unsigned integer, Least Significant Byte first
dword	32-bit integer, Least Significant Byte first
float	32-bit floating point
string	unsigned char (byte) array, ended with ‘0x1A’
char[]	unsigned char array with a given length
struct	data structure

Table 3.1*: Data types*

3.2.1 Internal Data Structures

Perhaps one of the most obvious data structures needed in Amore is the one for patient and diagnosis information. PatientDiagnosisData structure is given in table 3.2 as an example of numerous structure definitions in Amore. Size is always given in bytes.

Data description	Data type	Size
Name	**string**	≤ 100
Person ID	**string**	≤ 20
Age: “years.decimals”	**float**	4
Sex (0=male, 1=female)	**char**	1
Weight: “kg.decimals”	**float**	4
Height: “metres.centimetres”	**float**	4
Listening points (bitfield: bit 0 = Apex, bit 1 = Back, ...)	**dword**	4
Description of the “other” listeningpoint	**string**	≤ 50
Status of the patient	**string**	≤ 2000
Diagnosis	**string**	≤ 200
Creation date of the document	**string**	≤ 100
Creation place of the document	**string**	≤ 100
Author of the document	**string**	≤ 100
Read Only -protection flag (0 = read/write, 1 = read only)	**char**	1

Table 3.2*: PatientDiagnosisData structure*

3.2.2 File Formats

Several file formats are used by Amore for storing configuration information and temporary data. Most commonly used file format consists of a short, descriptive header, followed by a binary data structure. Length of the header is typically stored in the first byte as a single **char**. This kind of format is used for saving user preferences, display and processing data structures, filter definitions, tool definitions and database index data. These files are intended for internal use only.

More compatible format is selected for the document files; it complies with the RIFF standard. Document files consist of multiple data blocks, each having a standard RIFF header. File name extension used by Amore documents is .AMR. Document files can be opened with all programs complying with the RIFF standard; however, only the signal data block is generally supported and available in other applications, such as audio utilities for Windows . The blocks used in .AMR documents are presented in table 3.3.

Block description	Block name	Header size
Digital signal data (table 3.4)	WAVEfmt	44
Comment data (table 3.5)	WAVEcomm	31
Display setup data (table 3.6)	WAVEdisp	29
Patient information and diagnosis data (table 3.7)	WAVEinfd	29

***Table 3.3**: Structure of the .AMR document files*

Each .AMR document begins with the signal data block (table 3.4). This format is used by Windows audio waveform (.WAV) files.

Data description	Data Name	Data type	Size
"RIFF"	initials	**char[]**	4
Size of this RIFF header (typically 36)	hsize	**dword**	4
RIFF block name "WAVEfmt "	block name	**char[]**	8
Format block size (typically 16)	fsize	**dword**	4
Signal format (1 for Pulse Code Modulation)	sformat	**word**	2
Number of signal channels	channels	**word**	2
Signal sampling frequency (Hz)	sfreq	**dword**	4
Average bytes / second	avgbps	**dword**	4
Block alignment (bytes for one sample)	blockalign	**word**	2
Sampling resolution (bits / sample)	bits	**word**	2
"data"	dataInitials	**char[]**	4
Signal length (in bytes)	length	**dword**	4
i = LOOP length/blockalign, j = LOOP channels {			
sample; data type depends on the resolution: 8 bit samples: unsigned char (0...255) 16 bit samples: signed integer	sample[i][j]	**char or signed 16-bit integer**	max. 2
} j, i LOOP END			

Table 3.4*: Signal data block (the WAVEfmt block in .WAV files)*

The additional RIFF blocks are appended to the end of the document, right after the signal data. Detailed block structures are given in tables 3.5, 3.6 and 3.7.

Data description	Data Name	Data type	Size
"RIFF"	initials	**char[]**	4
Size of this RIFF header	hsize	**dword**	4
RIFF block name “WAVEcomm “	block name	**char[]**	8
Header sub block size (3 bytes in version 3)	subsize	**dword**	4
Comment data version number	version	**char**	1
Number of comments	comments	**word**	2
“data”	dataInitials	**char[]**	4
Data length (in bytes)	length	**dword**	4
i = LOOP comments {			
X-position (byte offset from the beginning)	xPos[i]	**dword**	4
Y-position (proportional placement)	yPos[i]	**dword**	
Associated view type	viewType[i]	**word**	
Associated signal channel	channel[i]	**word**	2
Length of the comment text (in bytes)	length[i]	**word**	2
Comment text	text[i]	**char[]**	length[i]
} i LOOP END			

Table 3.5*: Comment data block (WAVEcomm)*

Data description	Data Name	Data type	Size
"RIFF"	initials	**char[]**	4
Size of this RIFF header	hsize	**dword**	4
RIFF block name “WAVEdisp “	block name	**char[]**	8
Header sub block size (1 byte in version 1)	subsize	**dword**	4
Display data version number	version	**char**	1
“data”	dataInitials	**char[]**	4
Data length (in bytes, 10528 in version 1)	length	**dword**	4
DisplaySetupData structure	dData	**struct**	length

Table 3.6*: Display setup data block (WAVEdisp)*

Data description	Data Name	Data type	Size
"RIFF"	initials	**char[]**	4
Size of this RIFF header	hsize	**dword**	4
RIFF block name “WAVEinfd “	block name	**char[]**	8
Header sub block size (1 byte in version 1)	subsize	**dword**	4
Patient / diagnosis data version number	version	**char**	1
“data”	dataInitials	**char[]**	4
Data length (in bytes)	length	**dword**	4
PatientDiagnosisData structure	dData	**struct**	length

Table 3.7*: Patient information and diagnosis data block (WAVEinfd)*

3.3 User Interface

Amore project began as a research for an attractive and intuitive user interface which would help to increase the reliability and efficiency of heart murmur diagnosing. Informative and fast visualization was a key objective right from the beginning. In this sub-chapter, the visible results of the development work are presented and explained in broad outline, as an attempt to answer the question “What does it look like, and why?”. User interactions and the functionality of Amore are described more accurately in 3.4.2.

There are two basic operation states in Amore: monitoring and analysis. A typical monitoring display can be seen in figure 3.3: signal graphs and the level indicators (on the left side) are updated as new input is received. By default, graphs slide from right to left.

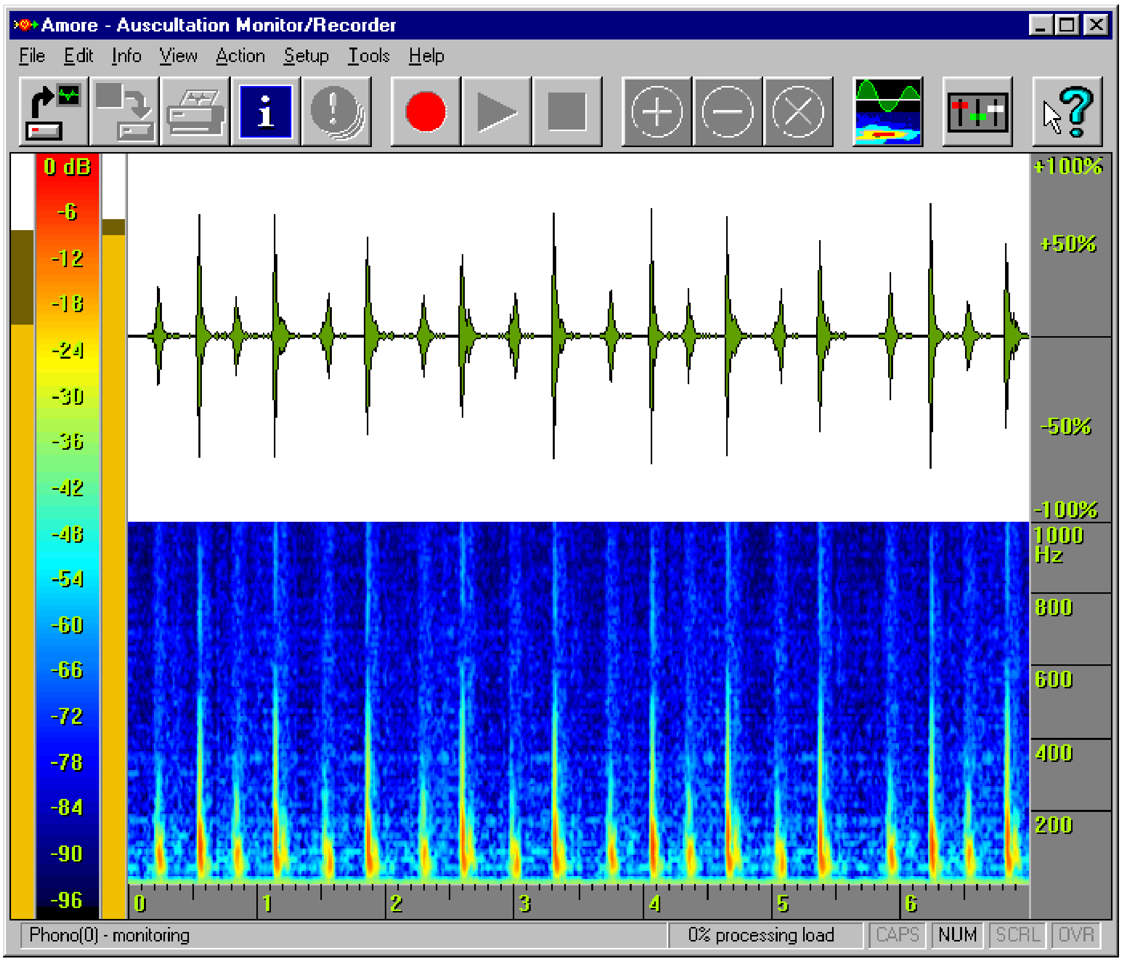


Figure 3.3*: Auscultation Monitor/Recorder v.4.7, currently monitoring input signal*

Record button is the one with a big, red, round spot on it. There are two alternative recording methods: either continuous recording (until stopped), or a “snapshot”. When the snapshot method is used, a circular buffer (size of which can be changed) is continuously filled with input signal. Display contents correspond to the snapshot buffer; in figure 3.3, a buffer of 7 seconds is used. The buffer is saved at any time by clicking the Record button.

Level indicators show the current signal amplitude in decibels (dB). Input levels are shown on the left side, output levels on the right side of the spectrogram color map. Level peaks are displayed with a darker color shade and with a slower attenuation rate.

The signal display consists of one ore more user defined views, each having one input signal channel associated with it. The graph type (amplitude or spectrogram) and appearance can be modified according to the needs and preferences of the user; appearance dialogs are presented in figures 3.5 and 3.6. Filtering and normalization of the displayed graphs can be done without altering the actual data. Also, the optional output channel for playback can be selected in a range that is supported by available hardware, i.e. the computer’s soundcard or a data acquisition board. Default display setup along with other user defined setups can be saved. Display Setup dialog is shown in figure 3.4. When the left mouse button is clicked, a selection/edit control appears at current position.

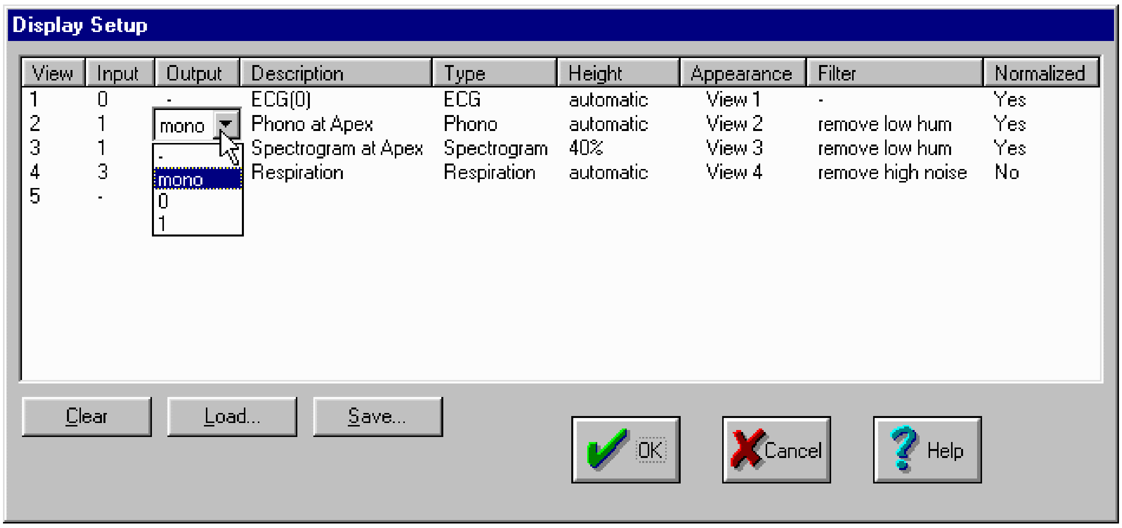


***Figure 3.4**: Display Setup dialog*

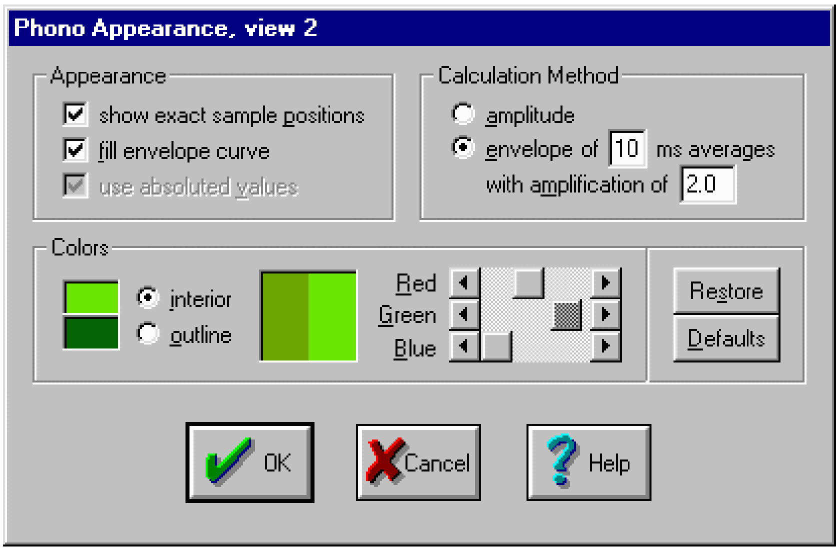


Figure 3.5: *Appearance dialog for amplitude type graphs; here, a Phono sub-type is selected for view 2*

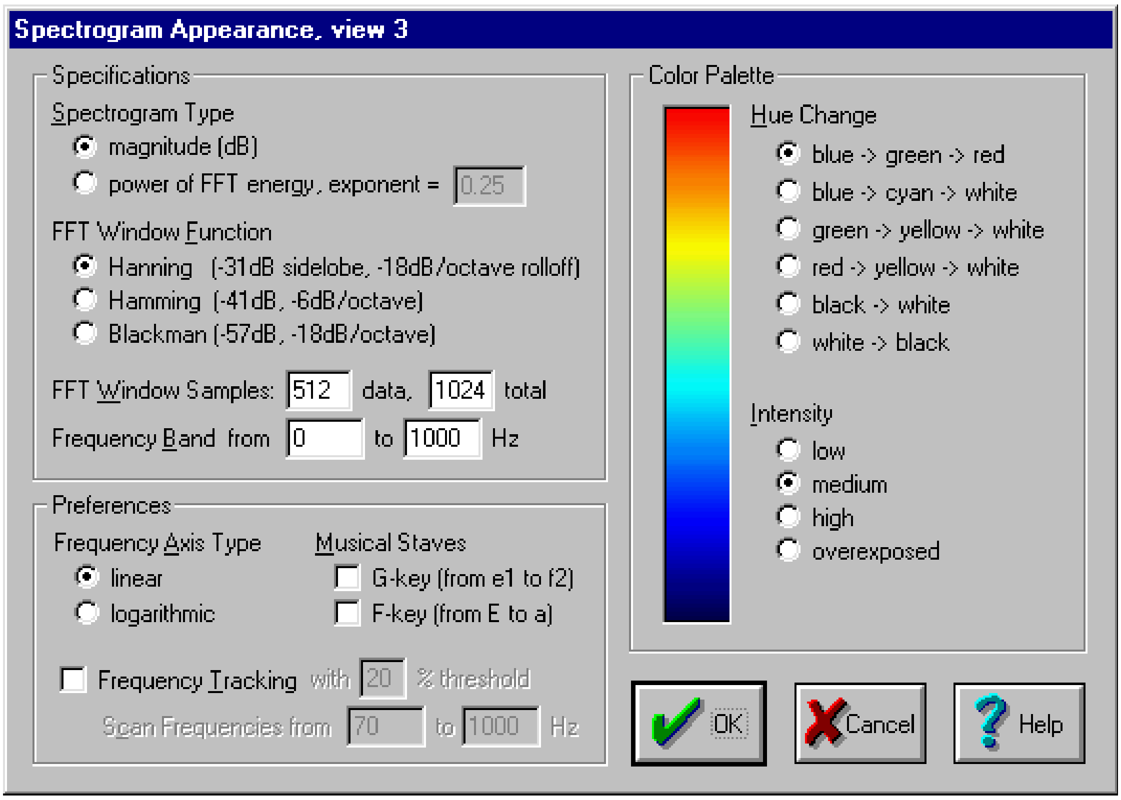


Figure 3.6: *Appearance dialog for spectrograms; default settings are selected for view 3*

By default, the hardware for signal input and output is automatically selected. Input and output devices can be selected independently. Available frequency and resolution range is automatically detected, as well as the maximum number of signal channels, and whether or not the selected hardware is capable of simultaneous input and output. Hardware Setup dialog is shown in figure 3.7.

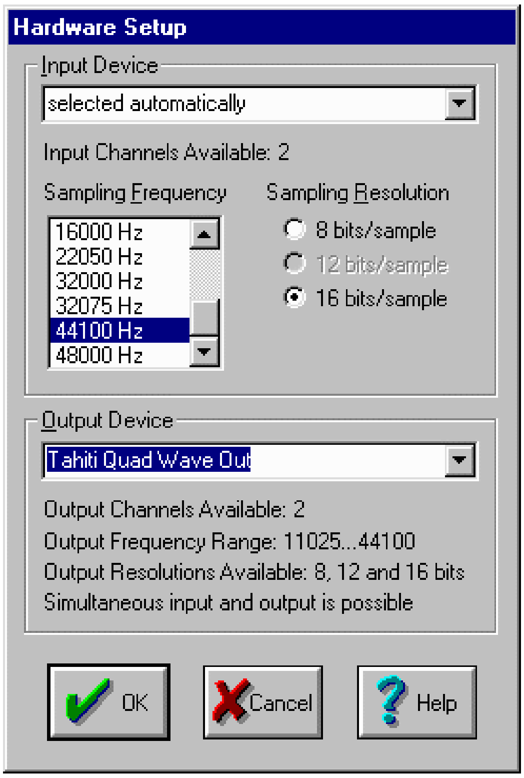


Figure 3.7: *Hardware Setup dialog*

Recording method, four monitoring options and the temporary file folder can be changed in the Preferences dialog (fig. 3.8), along with some other settings. When audio through monitoring is selected, both input and output level indicators are displayed during monitoring. This can be seen in fig. 3.3. Automatic zooming can be applied to newly recorded or opened signals. There is also an option for resetting the display setup and hardware to default values for all new documents.

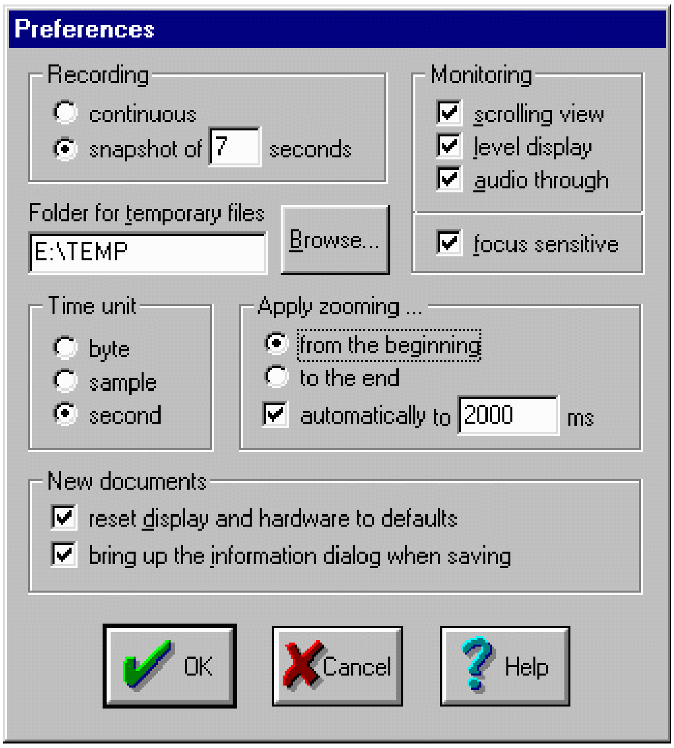


Figure 3.8: *Preferences dialog*

When a new signal is recorded, or a saved document is opened from the database, the appearance of Amore changes slightly (fig. 3.9):

- input monitoring is turned off and the input level indicator disappears
- a scroll bar appears at the bottom of the display
- grid lines for amplitude graphs appear
- Save, Print, Play and Zoom buttons become enabled
- for saved documents, the name of the document is displayed in the window's caption
- signal length is displayed in the status bar (at the bottom of the window) along with additional information related to the views, current selection, comments etc.
- when cursor is moved over the signal, labels are displayed on the horizontal and vertical axes, as well as in the intensity color map for spectrogram views

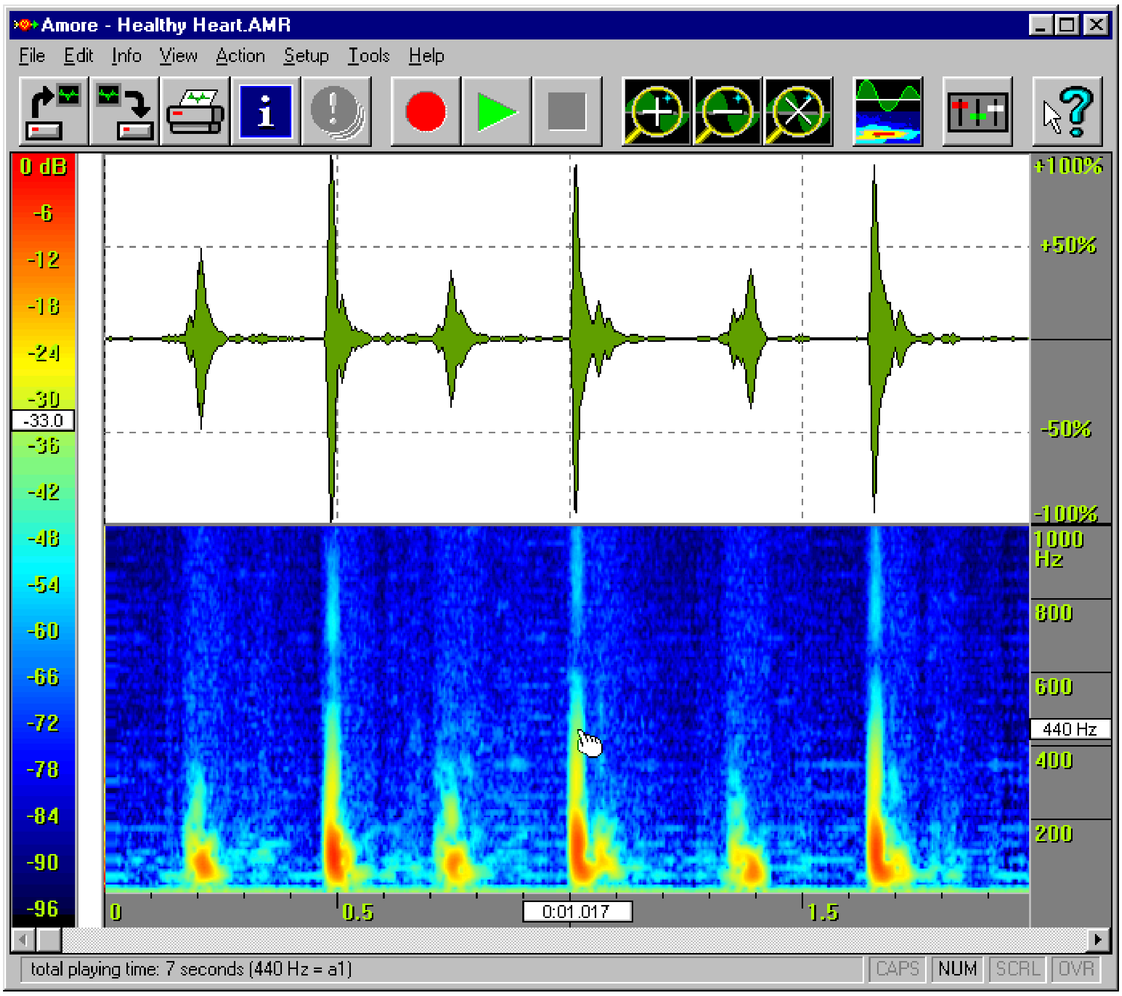


Figure 3.9: *Document "Healthy Heart.AMR" ready for analysis*

Figure 3.10 illustrates the area selection and measurement features of Amore. When the cursor is moved over the signal graphs while the left mouse button is held down, a measurement line is displayed, and additional information is displayed in the axis labels; in this case, the pulse seems to be ca. 90 bpm. The selection is displayed also after the mouse button is released. Comment marks (an exclamation mark inside a blue circle) can be added and positioned freely nearby interesting positions.

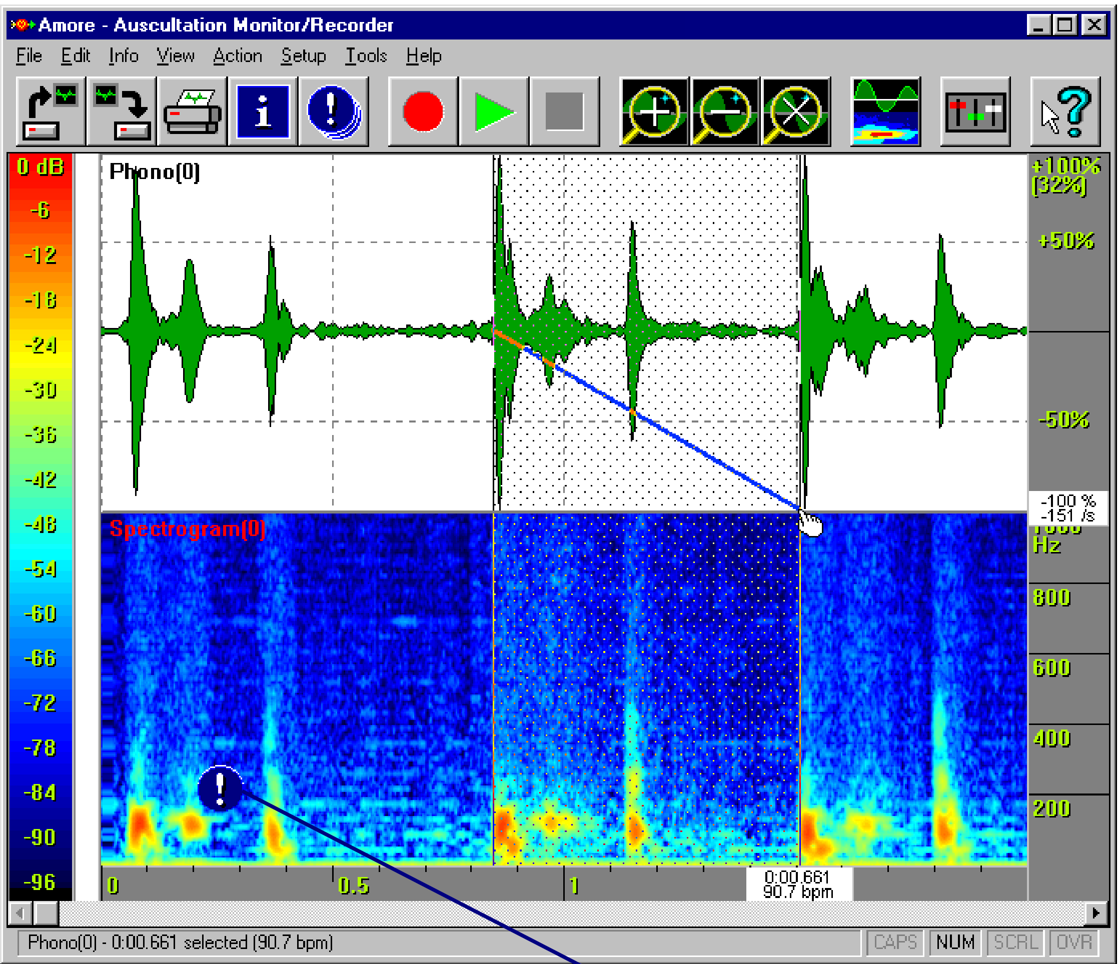


Figure 3.10*: Analysis of a biosignal*

Comment text is shown when the cursor moves over a comment mark. If a mark is clicked, a comment dialog (fig. 3.11) appears, with the associated comment.

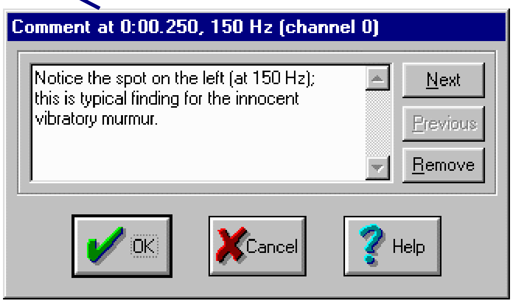


Figure 3.11*: Comment dialog*

It is important to fill in at least the most essential parts of the patient information for all new documents. This is why the Patient and Diagnosis Information dialog (fig. 3.12) is, by default, always displayed before new documents are saved. Some of the fields in the dialog are filled in automatically, either when the dialog is opened (Creation Date) or when the information can be derived from data in other fields (Age and Weight Index). There are three of four field groups in the dialog, depending on how you prefer to see it:

- patient information: name, person ID, age, sex, weight and height
- one or more listening points can be selected either by pressing the buttons or by selecting them from the list
- patient status and diagnosis; some standard diagnoses are given in the drop-down list; they can be edited and appended as is necessary
- When, Where and Who -information, and a switch to protect data as read-only

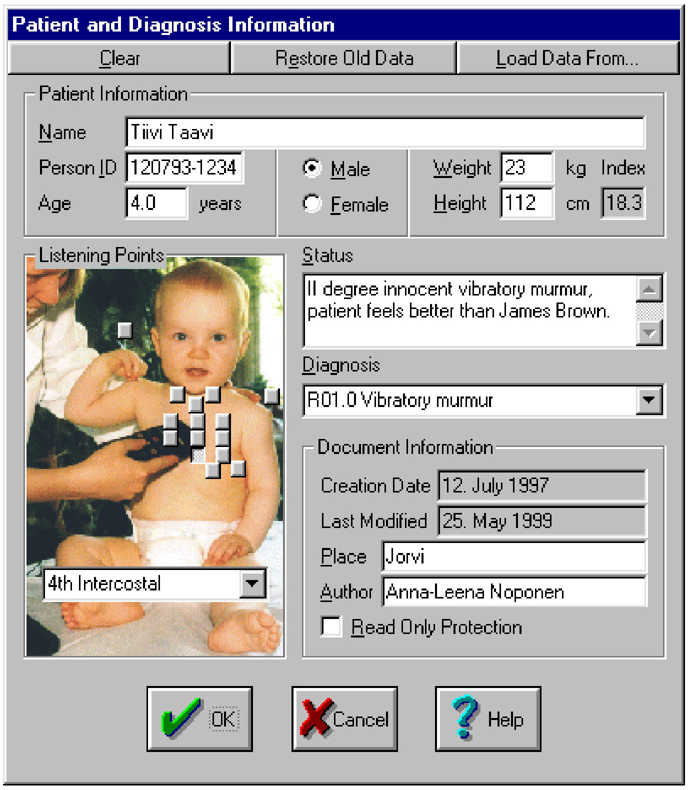


Figure 3.12: *Patient and Diagnosis Information*

Save As... dialog (fig. 3.13) is displayed when a new document is saved. A name suggestion is automatically generated.

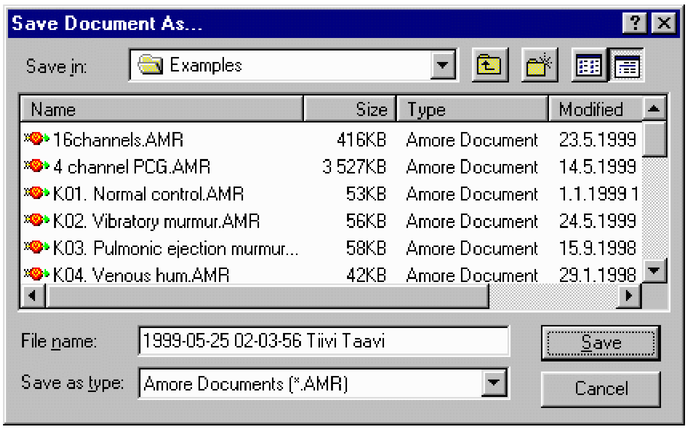


Figure 3.13: *Save As... dialog*

Documents are saved in a database, which is a group of folders on computer's harddisk. Database dialog (fig. 3.14) is the interface for managing databases and documents. When the File|Open... command is given, the database dialog appears and documents in the currently active database are displayed. Document list can be sorted by clicking the column headers (Name, Date, Diagnosis etc.) in a desired sorting priority order. In the upper left corner of the dialog, there's a drop-down list box for database selection. Currently, a database named "Examples" is active. Below the drop-down list, there are buttons for creating a new database folder and for selecting another folder on the harddisk as the root folder for databases. On the right of the database selection, there are buttons for importing a selected folder (fig. 3.15) as a new database, and exporting the current database to another location, such as a moveable disk. There are also import and export commands for single documents. Export Documents and Remove commands are enabled only if at least one document is selected in the list. However, if there are no documents in the list, Remove button is enabled for removing the database folder. Update button rebuilds the database index file. At the bottom of the dialog, either the Open or the OK button is displayed, depending on whether or not a document is selected in the list. Documents can be opened also with the Windows' Open File dialog by pressing Browse.

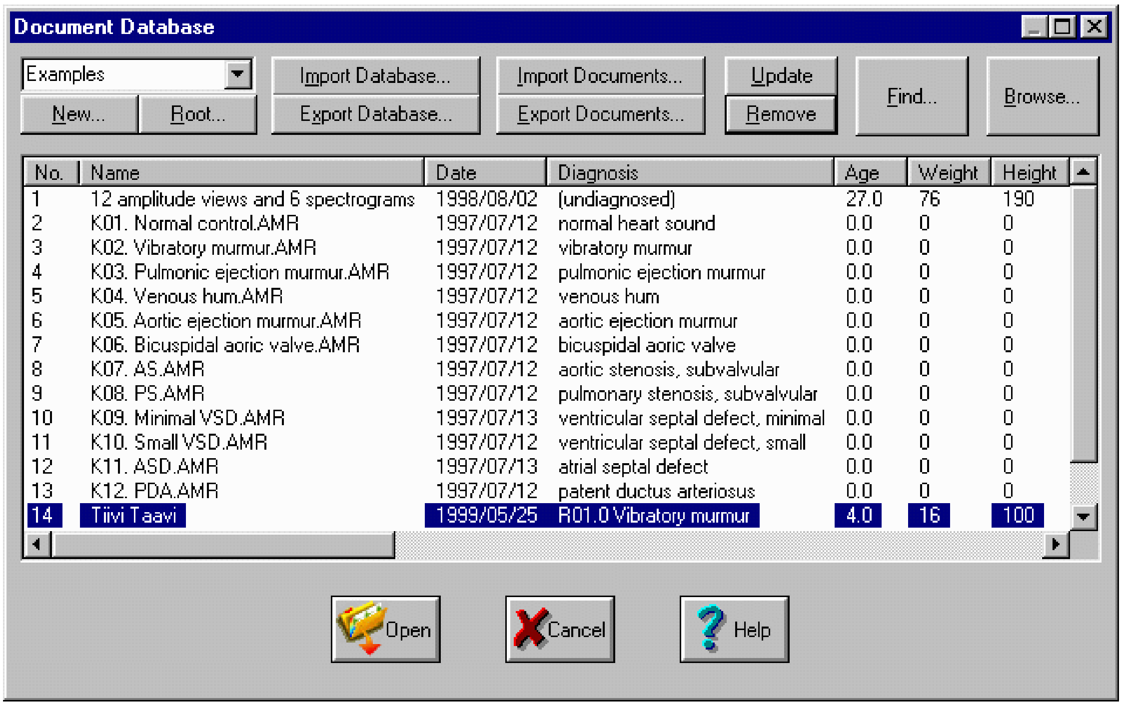


***Figure 3.14**: Database dialog*

Most of the buttons with an image are animated; for example, three bitmaps are used by the Open button in the Database dialog:

Normal appearance

Button is focused

Button is pressed

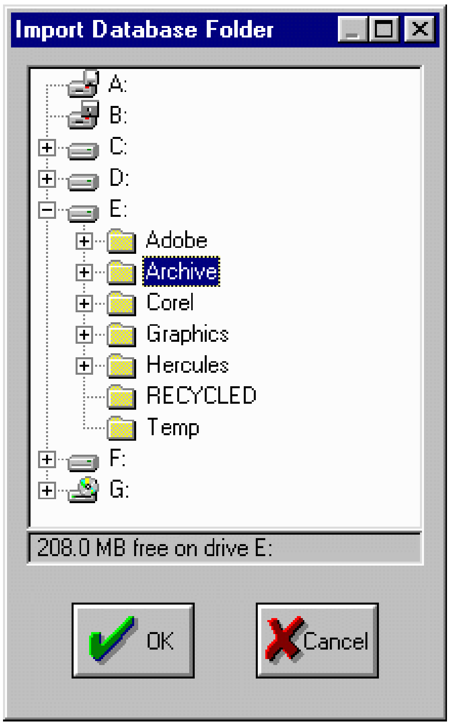


Figure 3.15*: Import Database*

As the number of documents and databases increases, it is necessary to have an efficient tool for finding documents. The Find button in the database dialog opens a tool dialog for searching documents matching with the given criteria (fig. 3.16). Since name, person ID and diagnosis contain different kind of data, a common search string is used. Search can be limited by giving lower and/or upper bounds for a selected variable, such as preson's age or total playing time of the signal. There is also an option for searching only males or females. When a match is found, it is highlighted in the document list, and the button text is changed to "Find Next". All matching documents can be copied or moved to a separate database; this way documents can be easily re-arranged.

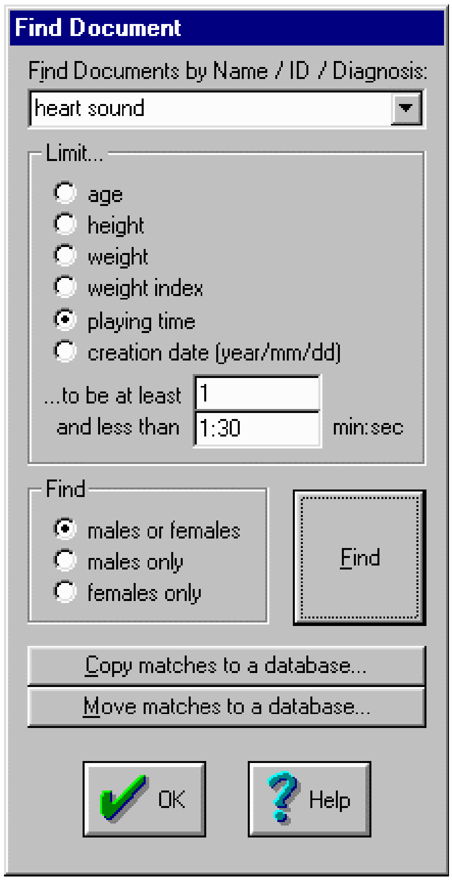


Figure 3.16*: Find Document*

Signals with up to 16 channels can be handled by Amore. New signal channels can be pasted into documents (fig. 3.17). In figure 3.18, a System and signal information dialog opened, while a four-channel signal is played in the background. Most soundcards have only two output channels; thus, channels must be muted or mixed. The play position indicator (inversed vertical line) is drawn only on views that have an output channel defined; in this case, views 0, 1 and 3.

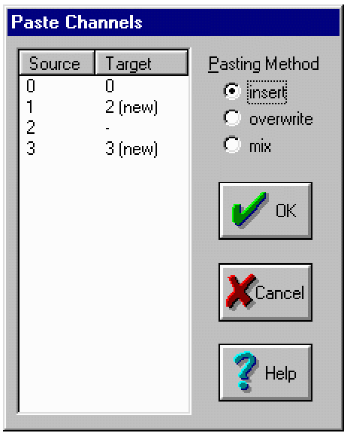


***Figure 3.17**: Paste dialog*

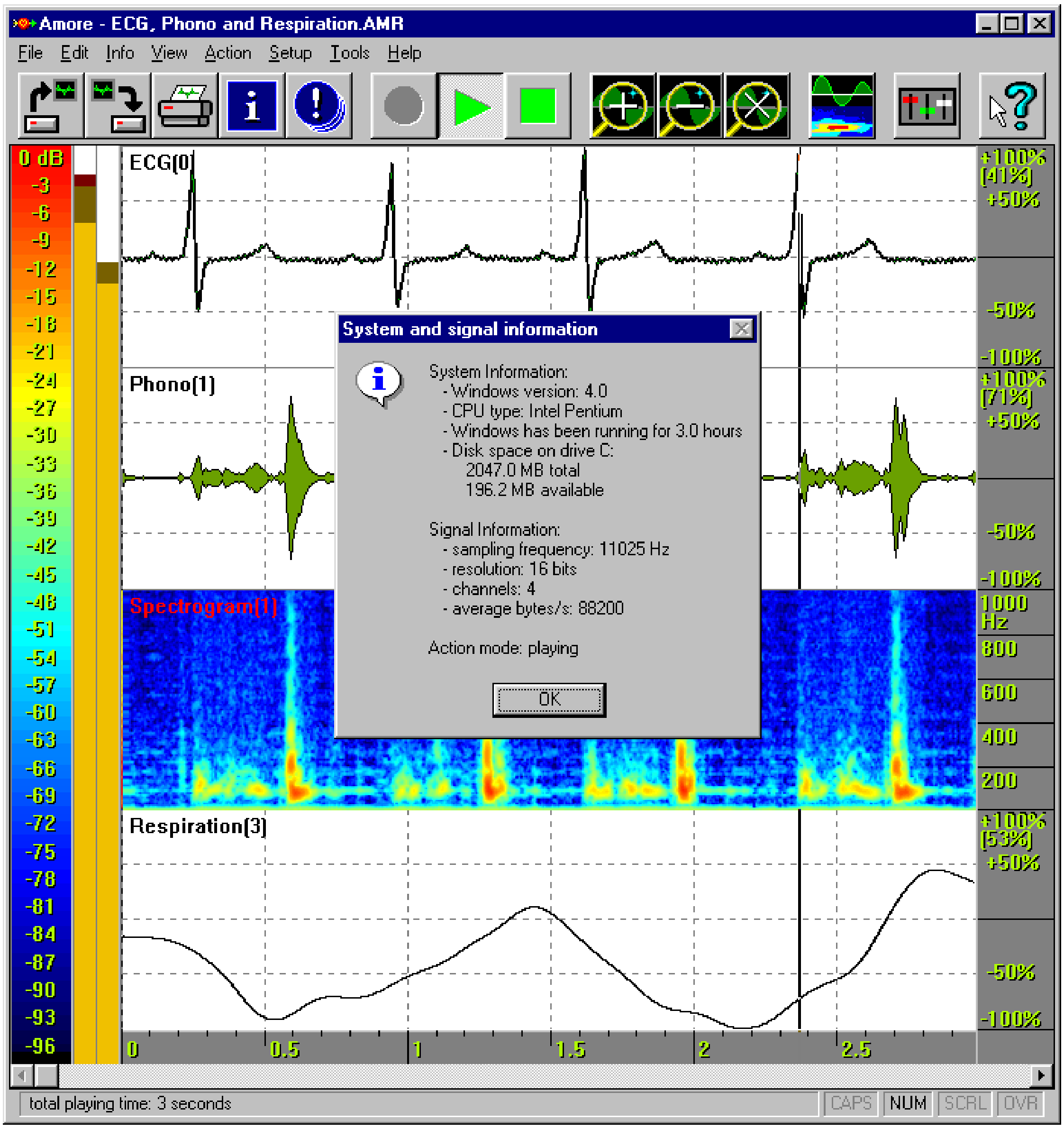


***Figure 3.18**: Multichannel playback. Output channels are mixed on the fly.*

Amore is capable of filtering the input signal with an user defined filter in real-time, within the limits of the calculation power provided by the computer. New filters can be created and the existing ones modified in the Filter Setup dialog (fig. 3.20). Calculation algorithms are implemented for lowpass, highpass, bandpass and bandstop IIR filters with either Butterworth or Chebyshev characteristics. The “Special” filter type provides a possibility to give the filter coefficients manually in the Filter Info dialog (fig. 3.21).

In early versions of Amore, only short FIR filters could be defined for input signals. Later on, also postprocessing functionality was added, with the Normalization feature for amplifying signals to use the full dynamic range. Most recently added feature is Delay for synchronization of multiple signal sources. Processing is defined separately for each input channel in the Pre- and Postprocessing dialogs (fig. 3.19). Definitions can be saved.

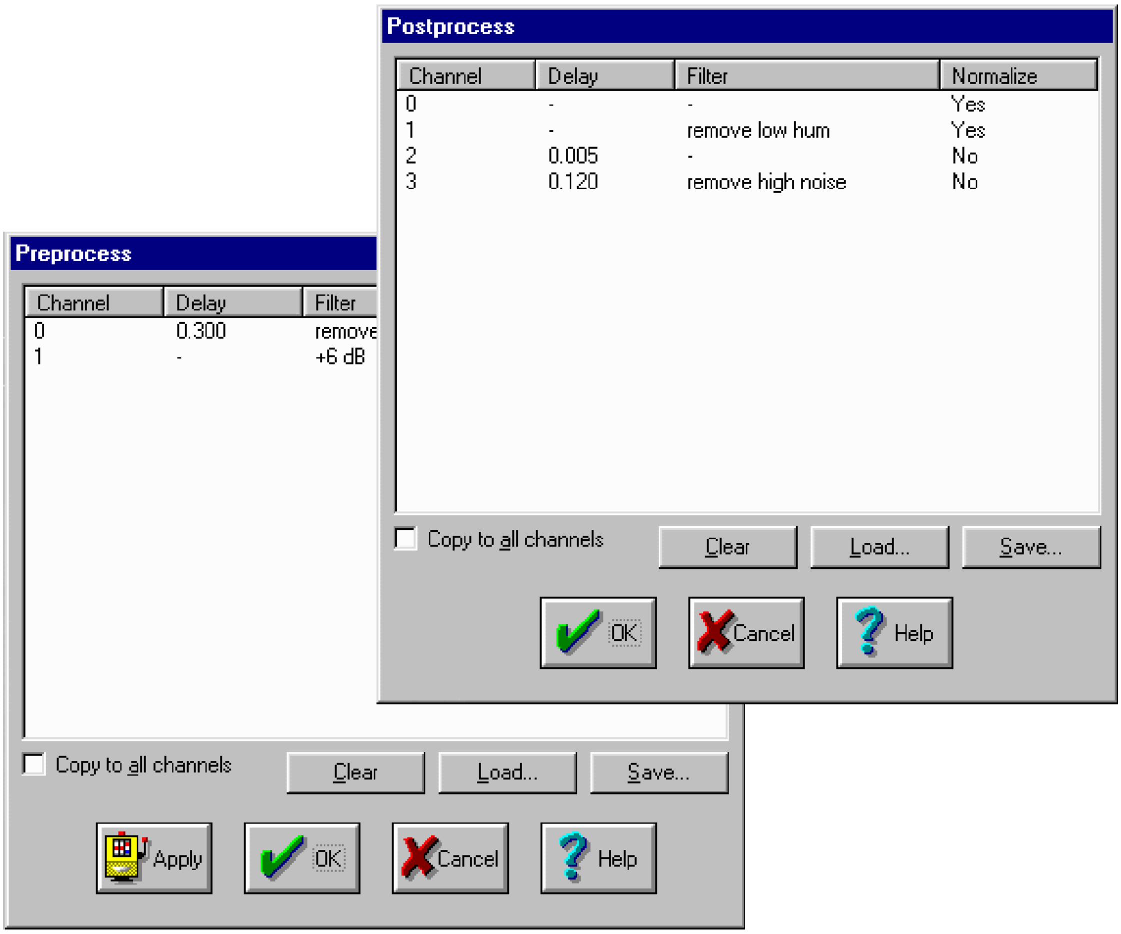


***Figure 3.19**: Pre- and Postprocess dialogs*

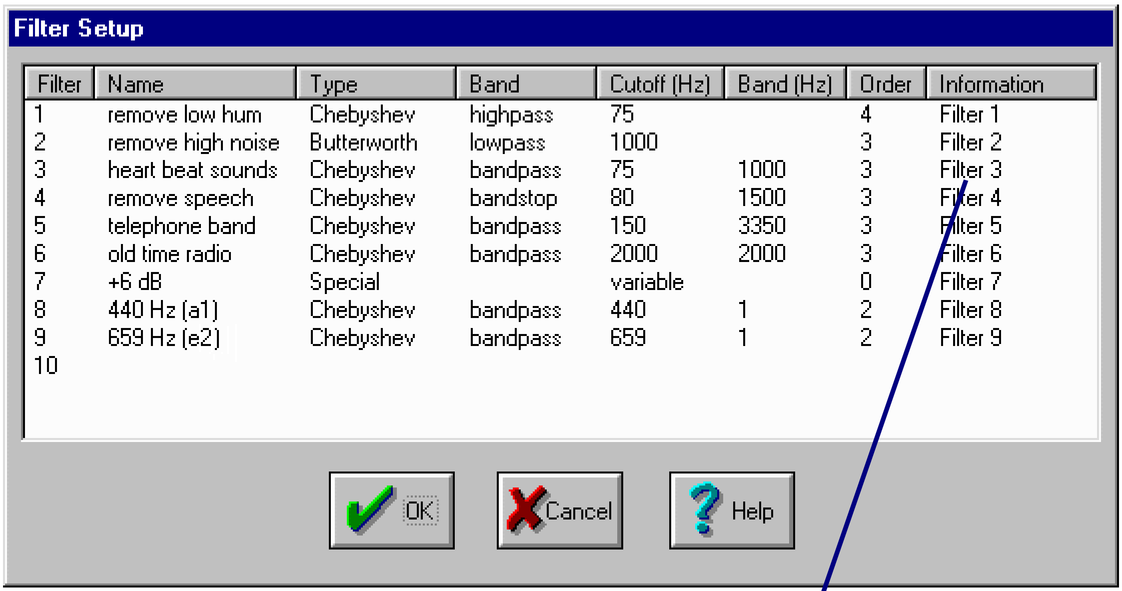


Figure 3.20: *Filter Setup dialog*

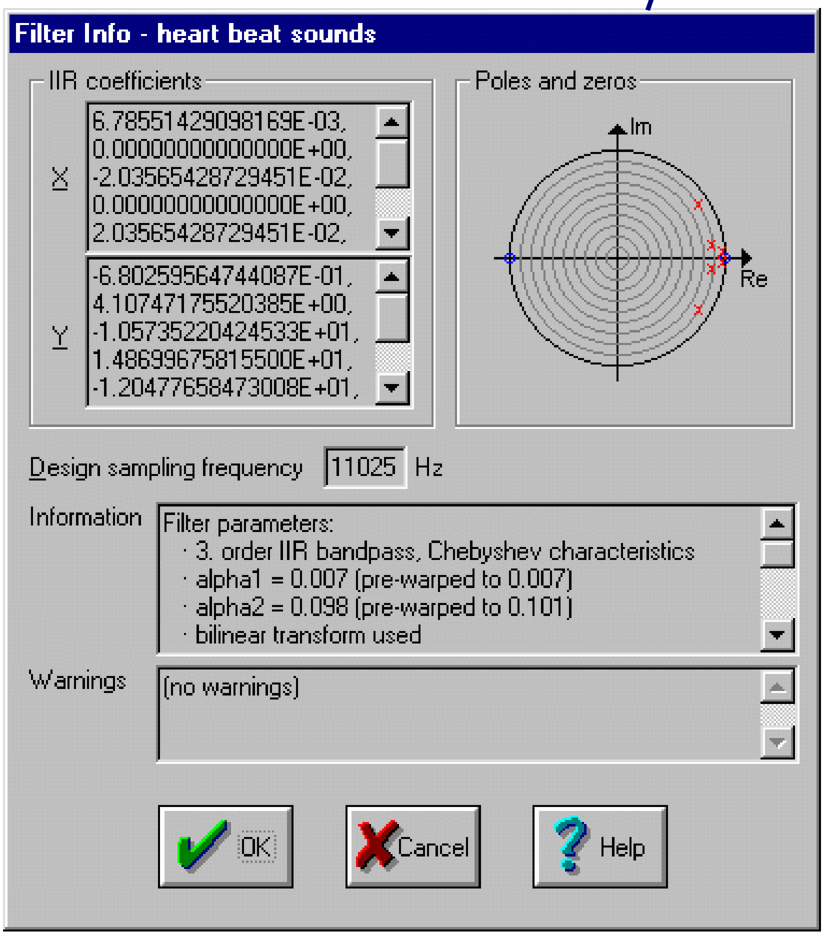


Figure 3.21: *Filter Info for the "heart beat sounds" bp filter; filter coefficients are automatically recalculated as the sampling frequency changes*

View properties and user interface colors can be easily modified. Sometimes just the background color may be a critical issue. An alternative front-end is given in figure 3.22:

- background, grid line and axis label colors have been changed from the default values in the Color Setup dialog (fig. 3.23)
- different color palettes are used for the two spectrogram views
- color intensity of spectrograms is a power function instead of dB magnitude; this is indicated also by the spectrogram color map numbers (percentages of max. value)
- left channel's spectrogram view has the frequency tracking property turned on; frequency with the highest intensity (10% threshold) is indicated with a white line (most recently tracked peak frequency is also displayed in the status bar)
- right channel's spectrogram view uses logarithmic frequency scale instead of linear
- more vertical space is used by spectrograms than by amplitude graphs (20% for both amplitude graphs, 30% for both spectrograms)
- view descriptions (Left/Right Channel) have been changed from defaults

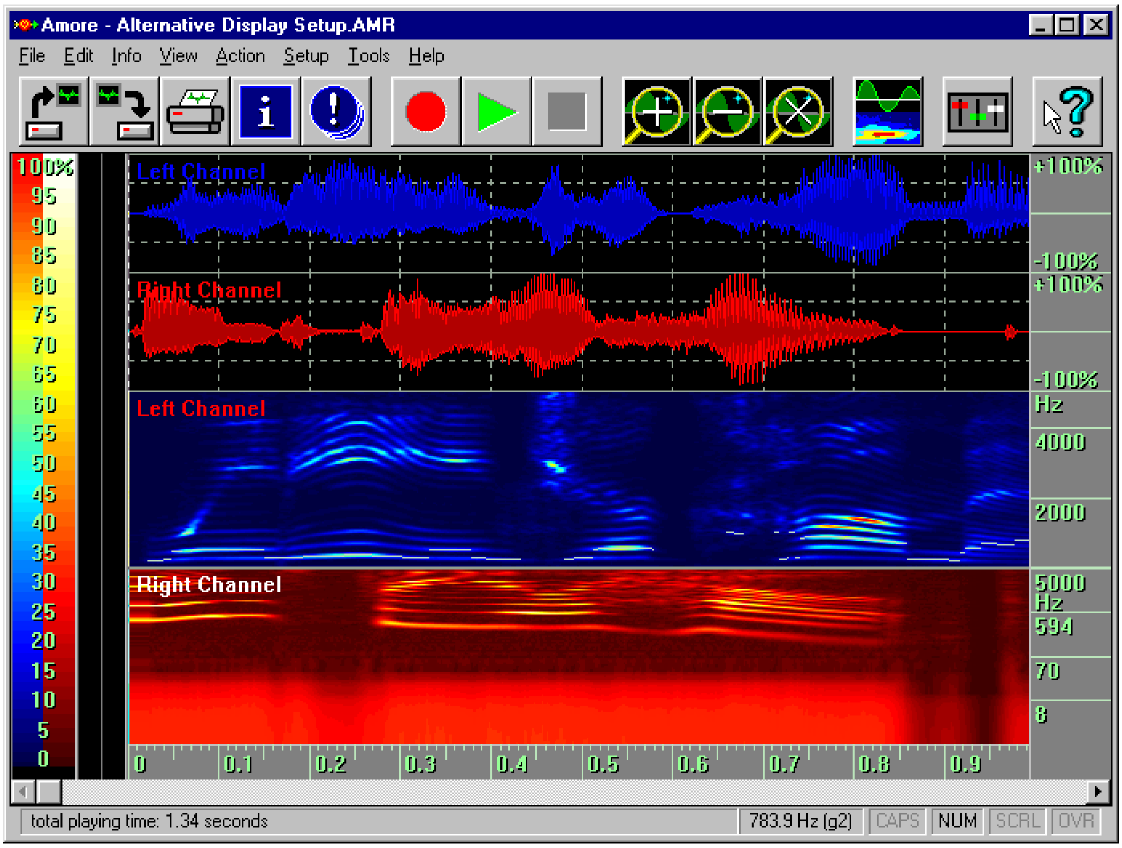


Figure 3.22*: An alternative display and color setup*

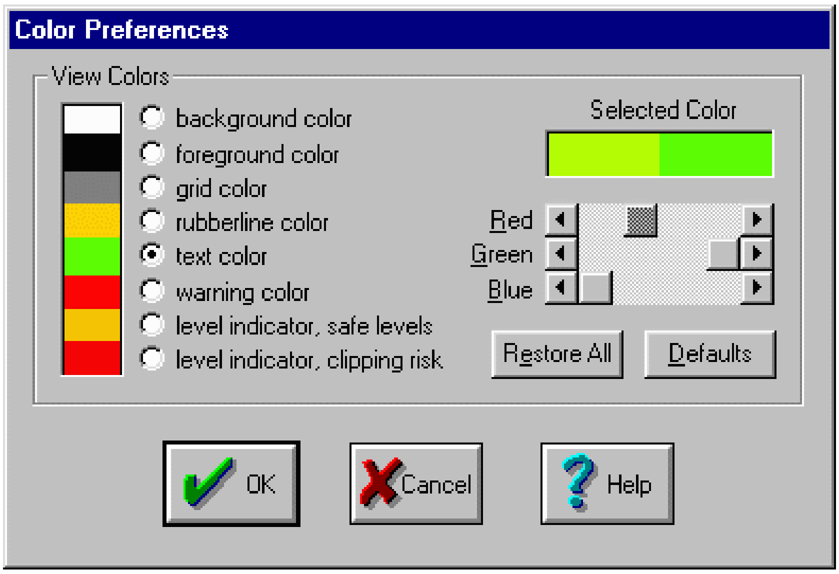


Figure 3.23*: Color Setup dialog*

Since the display is a shared hardware resource, palettized color modes may cause problems especially if several color intensive applications are run simultaneously. On the other hand, high color (15 or 16 bit) modes give only 5 or 6 bit depth for each primary color (red, green and blue) in comparison with the 8 bits provided by the palettized 256 color mode. The ultimate solution would be to use true color (24 or 32 bit) modes; however, so far these aren't generally available for very high resolutions (>1024 pixels). Amore supports both 256 color and high color modes.

Printing is implemented as "What You See Is Nearly What You Get": current display setup is used for formatting the output. Either color or grayscale palette is used, depending on the printer settings. If patient and diagnosis information is available, document is printed in the form of an auscultogram (fig. 3.25). During the print operation, a progress dialog is displayed (fig. 3.24). Printing can be stopped by pressing the Esc key.

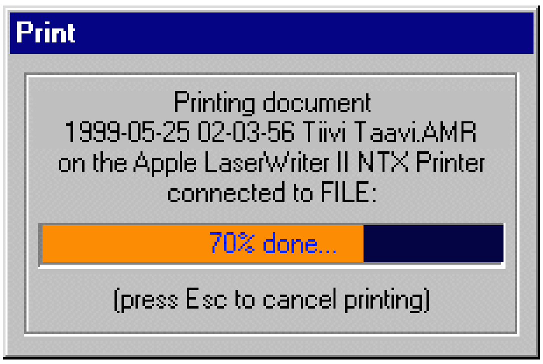


Figure 3.24*: Print progress*

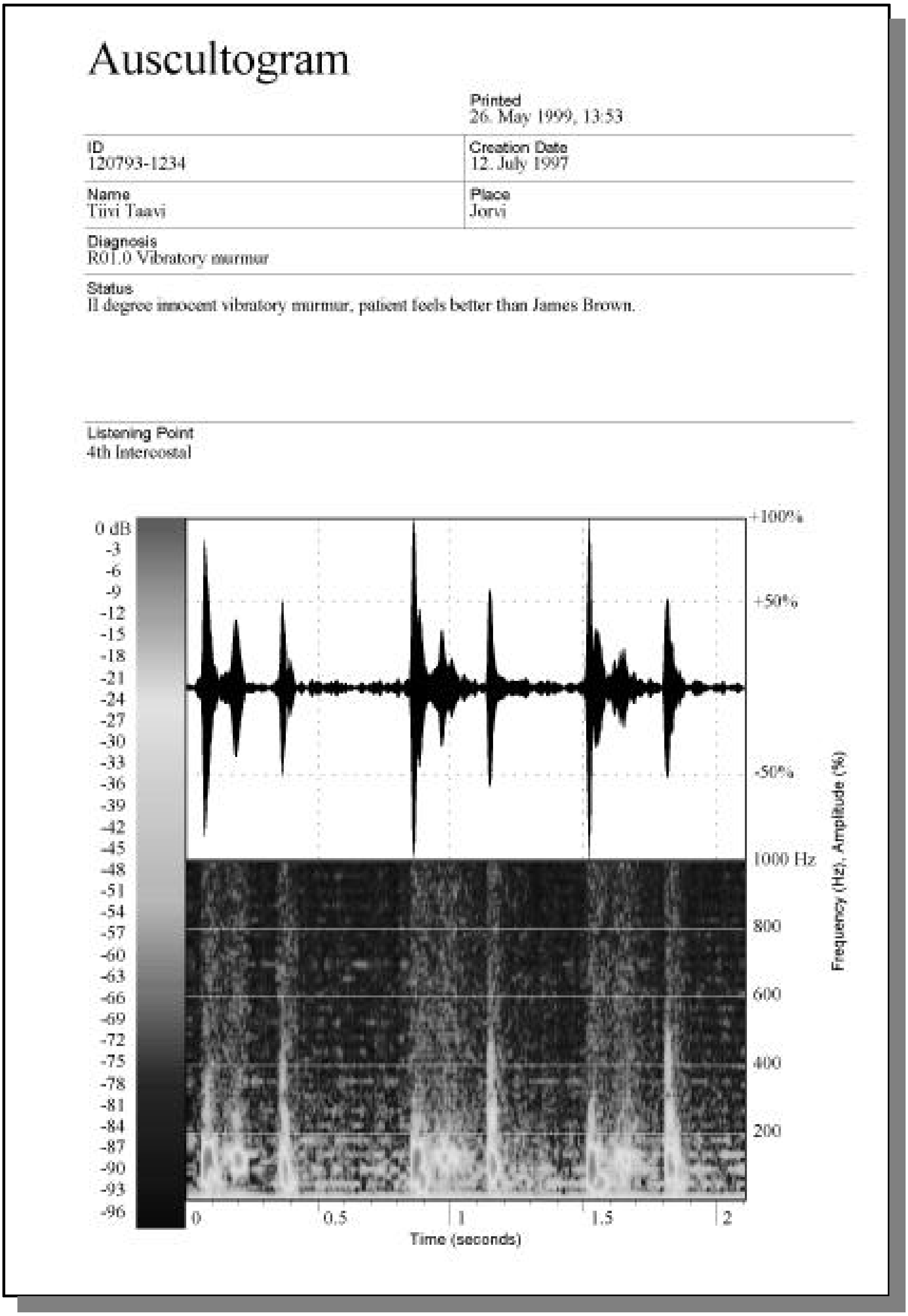


***Figure 3.25**: A grayscale auscultogram printout*

One of the first features in Amore was a button to launch the audio mixer application; it is important to adjust especially the signal input level of the soundcard correctly, so an easy access to the mixer is essential. Another useful feature is a possibility to send documents easily as e-mail attachments. This can done by starting a well-known web browser application with correct command-line parameters. There certainly are many other programs that could be used as tool applications for Amore; this is why the Tool Applications dialog (fig. 3.26) was implemented. Although this feature is unlikely used by less advanced users, it can greatly improve the efficiency of work. The e-mail application is automatically located in the harddisk and added to the tool application list when Amore is run for the first time after installation. Only descriptions of tools is shown in the menu.

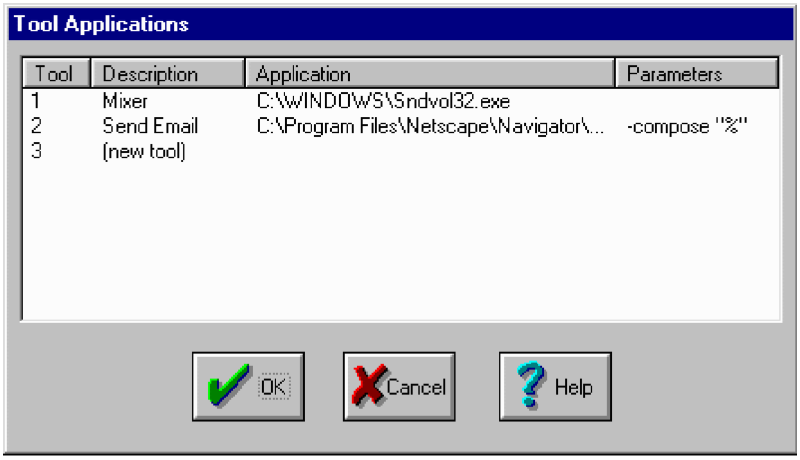


***Figure 3.26**: Tool Applications dialog*

Last but not least, there is the About Amore dialog box (fig. 3.27), containing version, registration and copyright information. The demonstration version included in the appendix CD-ROM should not be copied or made available via Internet; it is provided as a part of this master's thesis for evaluation only.

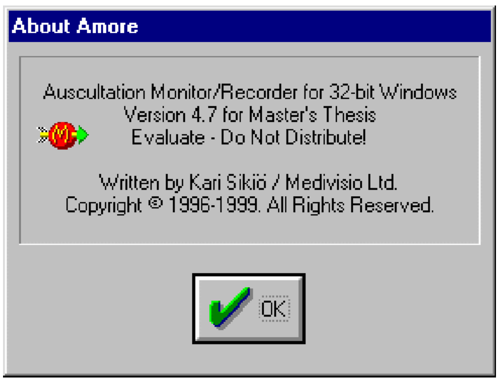


***Figure 3.27**: About Amore dialog*

It seems that only very few people actually read help texts - and even if they do, they've usually tried just about everything else (believe me, I have:) before going that far. Unfortunately, quite too often there is a good reason for this: reading helps doesn't help. This was kept in mind when Amore's help system was developed: the aim was to keep things simple, and include only the most essential basics clearly explained in the help text. Help is available in many forms: hint texts, point-and-click context sensitive help, and the traditional help hierarchy starting with help contents. Some help windows can be seen in figure 3.28. Green, underlined texts are links to other help topics.

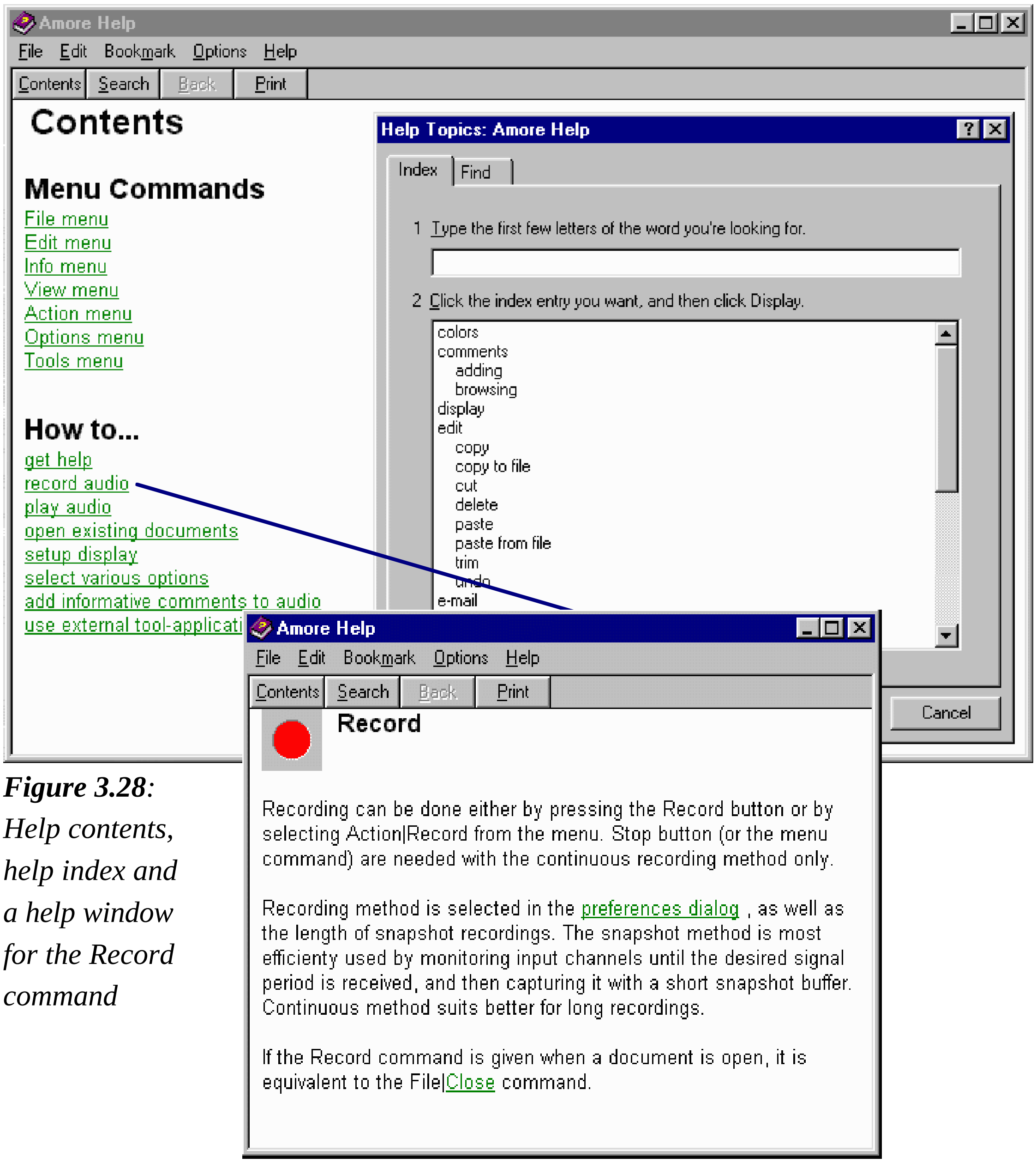


Figure 3.28: *Help contents, help index and a help window for the Record command*

3.4 Algorithms and Functions

The source code of Amore is property of Medivisio Ltd. There are more than 20000 lines of C++ source code (without the source files for OWL class implementations). Thus, only a very limited overview of the implementation of algorithms and functions can be presented here. Most of the functions and related user interactions are described in sub-chapter 3.4.2.

3.4.1 Algorithms

Although several intuitive algorithms have been created in this project, it would be out of the scope of this document to present them in a detailed level here. However, at least a detail level of pseudo-code representation is necessary for sufficient accuracy. In most time-critical implementations, it is essential how the possibilities of the programming language are used in order to optimize the execution. For these reasons I have decided to include only short examples of algorithmic solutions here, and to present them in their native form, C++ language, with thorough explanations.

Nowadays, even rather complex algorithms are implemented in the hardware as greater integration levels are achieved in the processor technology. As a consequence, some of the classical mathematical, highly optimized algorithm implementations are more and more often replaced with a call to hardware routines in the actual source code. One of these algorithms is certainly the Fast Fourier Transform (FFT). This is why I have chosen it to stand as an example of the source code. Both decimation in time and decimation in frequency -algorithms [6] are implemented, as well as the inverse FFT, using the decimation in time approach.

```
/* FFT shuffle index calculation. These indices give the bit-reversed
   order for real and imaginary array elements. Shuffle indices are
   calculated for all possible FFT window sizes (powers of 2).
   SV_MAX_FFTWINDOW is the maximum FFT window size = 2^SV_MAX_FFTBITS
   For example, if SV_MAX_FFTWINDOW = 4096, SV_MAX_FFTBITS = 12.
   M_PI = 3.14159265359. Initialize sin, cos and shuffleIndex tables
   to zero, for example by using the GPTR or GMEM_ZEROINIT flag with
   Win32 API's GlobalAlloc() function.
*/

char bit, bits=SV_MAX_FFTBITS;
short pos, half, windowSize=SV_MAX_FFTWINDOW;

// loop through all window sizes
for(; windowSize; windowSize>>=1,bits--) {
  // loop through the indices
  for(idx=0; idx<windowSize; idx++)
    // reverse index bits for ShuffleIndex
    for(bit=0; bit<bits; bit++) if(idx & (1<<bit))
      shuffleIndex[idx+windowSize-1]+=1<<(bits-1-bit);

  // sin & cos table calculation for decimation in frequency FFT
  half=windowSize>>1;
  pos=SV_MAX_FFTWINDOW-windowSize;
  for(idx=0; idx<half; idx++) {
    cosDIF[pos]=cos((double)idx*2*M_PI/windowSize);
    sinDIF[pos++]=-sin((double)idx*2*M_PI/windowSize); } }

// sin & cos table calculation for decimation in time FFT and IFFT
double Qreal, Qr, Qi, c, s;
pos=0;
for(short direction=-1; direction<=1; direction+=2)
  for(windowSize=1; windowSize<SV_MAX_FFTWINDOW; windowSize<<=1) {
    c=cos(M_PI/windowSize);           // cos(+-2*pi/FFT_window_size)
    s=sin(M_PI/windowSize)*direction; // sin(+-2*pi/FFT_window_size)
    cosDIT[pos]=Qr=1;
    sinDIT[pos++]=Qi=0;
    for(half=1; half<windowSize; half++) {
      // calculate Q^k; Qr+Qij = cos(angle)+sin(angle)*j
      Qreal=Qr;                       // save real part of Q
      Qr=Qr*c-Qi*s;                   // do complex multiplication
      Qi=Qi*c+Qreal*s;
      cosDIT[pos]=Qr;
      sinDIT[pos++]=Qi; } }
```

Algorithm 3.1: *Precalculations for the FFT algorithms*

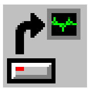
```
/* Fast Fourier Transform (FFT) algorithm is needed for spectrograms.
   "Size" is the FFT window's size. Transform is done in place, thus
   both input and output data is given in re[] and im[] tables.
   Input data for the decimation in time FFT and output data of the
   decimation in frequency FFT is in bit-reversed order.
   Radix-2 butterfly calculation method is used here.
   word is 16-bit unsigned interger, long is 32-bit signed integer.
*/
void SoundView::DoFFT(word size, long *re, long *im,
                      bool forward, bool decimationInTime)
{
  short angle;                          // index to sin & cos tables
  word idx, count;                      // loop indexes & counters
  register word idx1, idx2, jump;       // - " -
  register long real, imag;             // variables for butterfly
  register float cs, sn;
  // decimation in time -algorithm (both FFT and IFFT supported)
  if(decimationInTime) {
    angle=forward?0:SV_MAX_FFTWINDOW-1;
    for(idx=1; idx<size; idx<<=1)
      for(jump=idx<<1,count=0; count<idx; count++) {
        // precalculated sin & cos tables speed up things remarkably
        cs=cosDIT[angle];
        sn=sinDIT[angle++];
        // radix-2 butterfly-loop
        for(idx1=count; idx1<size; idx1+=jump) {
          idx2=idx1+idx;                // perform 2-point DFT
          real=re[idx2]*cs-im[idx2]*sn; // size*log2(size) times
          imag=im[idx2]*cs+re[idx2]*sn;
          re[idx2]=re[idx1]-real;
          im[idx2]=im[idx1]-imag;
          re[idx1]+=real;
          im[idx1]+=imag; } } }

  // decimation in frequency -algorithm;
  // only forward FFT is supported so far
  else {
    for(angle=SV_MAX_FFTWINDOW-(idx=size); idx>1;)
      for(jump=idx,idx>>=1,count=0; count<idx; count++) {
        cs=cosDIF[angle];
        sn=sinDIF[angle++];
        for(idx1=count; idx1<size; idx1+=jump) {
          idx2=idx1+idx;
          real=re[idx1]-re[idx2];
          imag=im[idx1]-im[idx2];
          re[idx1]+=re[idx2];
          im[idx1]+=im[idx2];
          re[idx2]=cs*real-sn*imag;
          im[idx2]=sn*real+cs*imag; } } }

  // normalization (re[],im[]/=size) for inverse FFT
  // is combined with other scaling and done elsewhere
}
```

Algorithm 3.2: *FFT algorithm implementations*

3.4.2 Functions

 **Open document**

Purpose: open an existing document from a database.

User input:

- Open button (toolbar ) or File|Open... command (menu)
- In the Open Document dialog:
  - database list for database selection
  - New... button = create a new database to the database collection
  - Root... button = change to other database collection (change root folder)
  - Import Database... button = import a new database to the database collection
  - Export Database... button = export current database to a specified location
  - Import Documents... button = import document(s) to current database
  - Export Documents... button = export selected document(s) to another database
  - Update button = re-read the hard disk folder to update the database index file
  - Remove button = remove selected document(s) or current database if it's empty
  - Find... button = find a document using the Find Document dialog
  - Browse... button = open a single file by browsing the harddisk folders
  - Ok button = close the dialog and open the selected file
  - Cancel button = close the dialog without opening a file
  - Help button = display help on using the Open Document dialog

Description:

The documents are saved in a database consisting of a set of database directories. Each directory contains a set of documents. The contents of the most recently used database are shown by default, but any database in the database list can be selected. A single document can be searched for by filling in one or more fields in the search limit dialogs.

Single documents or whole database directories can be imported from and exported to a specified location. New databases can be created by filling in a name that is not used by any database to the database selection.

Outputs:

The effects of the import operations or database selection are updated immediately to the document list. When Ok button is pressed (either in the Open Document dialog or in the Browse dialog), the selected file is opened and its contents are displayed.

Exception handling:

When insufficient disk space or other problems are detected, user is informed.

Reopen document

Purpose: discard all changes made to the contents of this document.

User input:

- File|Reopen command (menu)

Description:

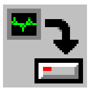
Especially when experimenting with signal processing tools and the display setups, it's often useful to reset the situation and reopen the document. This is the fast way to do it.

Outputs:

See Open document.

Exception handling:

If an error occur while taking a backup of the document (which is done always before the signal contents change), user is informed.

 **Save document**

Purpose: save currently open document and associated data to the current database.

User input:

- Save button (toolbar ) or File|Save command (menu)

Description:

Documents are saved in the currently active database. An unique name suggestion is generated for each new document, consisting of current date and time, the patient's name and, if necessary, a number to separate it from another document with an identical name. Documents have ".AMR" extension by default, but can also be saved as ".WAV".

Outputs:

If no patient information has been given for the document, the Patient and Diagnosis Information dialog is displayed. At least the patient's name should be filled in before the document is saved. When a new document is saved, the file name is displayed in the window caption. Also, a message sound is played to indicate a successful save operation.

Exception handling:

When insufficient disk space or other problems are detected, user is informed.

Save document As...

Purpose: save currently open document and associated data to a specified location.

User input:

- File|Save As... command (menu) = open the Save As... dialog

- In the Save As... dialog:
  - Ok button = save the document with the selected name and close the dialog
  - Cancel button = close the dialog without saving the document
  - Help button = display help on using the Save As... dialog

Description:

Default name is generated for new documents (see Save document). Documents that are saved outside the database can't be located with the Find command in Database dialog.

Outputs:

Identical with the Save document -command.

Close document

Purpose: close currently open document and clear memory for a new recording.

User input:

- File|Close command (menu)

Description:

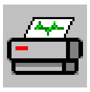
If the document has changed since last save operation, user is asked if the changes should be saved before the document is closed.

Outputs:

After the document is closed, the input signal monitoring is activated on the graphs.

Exception handling:

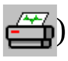
If user wants the document to be saved before it is closed, and the save operation fails, an error message is displayed and the close operation is canceled.



Print document

Purpose: print document and associated patient information with the specified printing device.

User input:

- Print button (toolbar ) or File|Print command (menu) to open the Print dialog

Description:

The printed signal graphs resemble the currently displayed graphs as much as possible, depending on the printer properties. If diagnosis information is given, it will be printed with the signal graphs.

Outputs:

While printing, a progress bar and a Cancel button are shown to enable canceling of the print operation.

Exception handling:

Windows will inform user if there are problems with the printing device.

Page Setup

Purpose: define the printout properties, such as paper dimensions and orientation.

User input:

- File|Page Setup... command (menu) = open the Page Setup dialog

Description:

The standard Windows Page Setup dialog is displayed here. Contents of the dialog vary depending on the Windows version; however, a Help button is always included.

Outputs:

The settings in this dialog affect the printer output.

Exception handling:

Windows will inform user if there are problems with the printing device.

Exit

Purpose: close the document and exit the program.

User input:

- File|Exit command (menu)

Description:

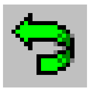
If the document has changed since last save operation, user is asked if the changes should be saved before closing it and leaving the software.

Outputs:

The software window is closed.

Exception handling:

If user requests the document to be saved before exit, and the save operation fails, and error message is displayed and the exit command is canceled.

 **Undo**

Purpose: undo most recent edit operation.

User input:

- Edit|Undo command (menu)

Description:

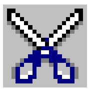
This applies to text edit controls only. Original signal contents can be restored with File|Reopen command.

Outputs:

The previous text in the edit control is restored.

Exception handling:

-

 **Cut**

Purpose: cut selection to the clipboard.

User input:

- Edit|Cut command (menu)

Description:

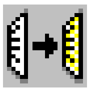
Selected data is moved to the Windows clipboard. This applies both to text edit controls and signal data.

Outputs:

During the cut operation, the cursor shape changes to indicate that user should wait for the operation to complete. When finished, the selection is removed and the effect of the cut operation is shown immediately.

Exception handling:

If the data is too large to fit in the clipboard, or other problems are encountered, user is informed and the cut operation is canceled.

 **Copy**

Purpose: copy selection to the clipboard

User input:

- Edit|Copy command (menu)

Description:

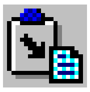
Selected data is copied to the Windows clipboard. This applies both to text edit controls and signal data.

Outputs:

During the copy operation, the cursor shape changes to indicate that user should wait for the operation to complete. When finished successfully, a message sound is played.

Exception handling:

If the data is too large to fit in the clipboard, or other problems are encountered, user is informed and the copy operation is canceled.

 **Paste**

Purpose: paste the clipboard contents to the current position.

User input:

- Edit|Paste command (menu)
- In Paste dialog (if signal data is pasted)
  - channel list view for defining the channelization
  - paste method radiobuttons to select between insert, overwrite or mix pasting
  - Ok, Cancel and Help buttons

Description:

This applies both to text edit controls and signal data. Clipboard contents are copied to current position, indicated with cursor in text edit controls and with position indicator in signal graphs. With signal data, the optional target channel is freely selected for each source channel in a Paste dialog. There are three paste modes available:

- insertion (default): existing data is slided forward to make room for pasted data
- overwrite: existing data is overwritten
- mix: existing data is mixed with the pasted data where overlapping occurs

Outputs:

During the paste operation, the cursor shape changes to indicate that user should wait for the operation to complete. When finished successfully, a message sound is played.

Exception handling:

If insufficient disk space or other problems are encountered, user is informed and the paste operation is canceled.

Copy to file

Purpose: copy selected signal data to a file

User input:

- Edit|Copy to file... command (menu)

Description:

Selected signal data (or whole signal if no selection is currently done) is copied to a file. The data that is associated with the selection is written to the file as well. File name is provided by user.

Outputs:

The standard Windows File Save As... dialog is displayed to let the user give a name for the target file. During the copy operation, the cursor shape changes to indicate that user should wait for the operation to complete. When finished successfully, a message sound is played.

Exception handling:

If insufficient disk space or other problems are encountered, user is informed and the copy operation is canceled.

Paste from File

Purpose: paste the contents of a file to the current position.

User input:

- Edit|Paste from File... command (menu)

Description:

The signal contents of the selected file are copied to current position, indicated with the position indicator in the signal graph. The user may select channelization and paste mode in the Paste dialog (see Paste). If both the input file and currently open document have diagnosis information, user is asked if the existing information should be replaced with that of the source file.

Outputs:

The standard Windows File Open... dialog is displayed to let the user select the source file. During the paste operation, the cursor shape changes to indicate that user should wait for the operation to complete. When finished successfully, a message sound is played and the result can be seen in the updated signal graphs.

Exception handling:

If insufficient disk space or other problems are encountered, user is informed and the paste operation is canceled.

Delete

Purpose: delete signal data.

User input:

- Edit|Delete command (menu)

Description:

Selected portion of the signal is deleted. If a selection hasn't been defined, the document is closed without saving.

Outputs:

During the delete operation, the cursor shape changes to indicate that user should wait for the operation to complete. When finished, the selection is removed and the effect of the delete operation can be seen in the updated signal graphs. On success, a message sound is also played.

Exception handling:

If any problems are encountered, user is informed and the delete operation is canceled.

Trim

Purpose: delete signal data outside the selection.

User input:

- Edit|Trim command (menu)

Description:

Signal data outside the current selection, is removed.

Outputs:

During the trim operation, the cursor shape changes to indicate that user should wait for the operation to complete. When finished, the selection is removed and the effect of the delete operation is shown immediately in the signal graphs. On success, the message beep is also played.

Exception handling:

If any problems are encountered, user is informed and the trim operation is canceled.

Diagnosis Information

Purpose: serve as a data sheet for patient and diagnosis information.

User input:

- Diagnosis Information button (toolbar) or Info|Diagnosis... command (menu)
- In the Diagnosis Information dialog
  - patient information fields (Name, Person ID, Age, Sex, Weight and Height)
  - listening point selection radiobuttons
  - status field
  - diagnosis field
  - document information fields (Creation Place, Author and Read Only checkbox)
  - Clear All button for clearing all fields
  - Restore Old Data button to use the data of the previously open document
  - Load Data From... button to copy the data from a selected document
  - Ok, Cancel and Help buttons

Description:

It is up to user to select which fields of the Diagnosis Information dialog are filled, and how accurately. All fields can be left empty. However, it would be recommended to fill in at least the patient name, person ID and the listening point at which the signal is saved. All fields can be filled in afterwards, and especially with diagnosis fields this may be the most usual case. Any diagnosis may be written into diagnosis combobox, but it is recommended to use one of the pre-defined, standardized diagnoses and append them with more accurate classification when necessary.

Input fields of the dialog are cleared for new recordings. When several listening points are studied from the same patient, the cleared patient data fields can be quickly restored with the Restore Old Data button. Pressing the Load Data From... opens the Database dialog for selecting a document whose patient and diagnosis information is to be copied.

Outputs:

When the person ID is given in form “ddmmyy” (day, month, year; two digits for each), age is calculated and displayed with one decimal accuracy if the field is empty. When both weight and height are given, the weight index (weight (kg) / length (m) ^{2}) is calculated. Creation and modification dates of the document are displayed in the document information -group. If the Read Only option is selected, all modifications are disabled when the document is opened next time, until the protection is turned off.

Exception handling:

The Load Data From...command: if the selected document can’t be found, is corrupted or contains no patient/diagnosis information, user is informed.

System and signal information

Purpose: display information on the signal parameters and system status.

User input:

- Info|System and signal... command (menu)

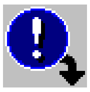
Description:

This message dialog contains following information:

- Windows version number (4.0 for Windows’95)
- total and available hard disk space on the currently used drive
- playing time of the signal (if a document is currently open)
- signal parameters (such as sampling frequency and resolution)
- current action mode (monitoring, recording, playing etc.)

Outputs:

The dialog.



Add Comment

Purpose: attach a textual comment to document's signal data.

User input:

- Add Comment Here! command in the floating pop-up menu
- In the Comment box
  - comment text edit control
  - Next and Previous buttons to change to the next or the previous comment
  - Remove button to remove current comment
  - Ok, Cancel and Help buttons

Description:

A comment is a short, descriptive text that is attached to a specified position and channel in the signal data. This is done by moving the mouse cursor to the intended comment position and clicking the right mouse button; a floating pop-up-menu is activated, in which the Add Comment Here command can be selected. A small dialog appears for writing the comment text. Once a command is created, it can be moved to another position by pressing the left mouse button down over a comment mark, and dragging (moving the mouse while the left button is held down) the comment onto a new position.

Outputs:

A comment mark is shown on all views of same type that share the same signal channel that the comment is associated with. Comment data are saved with the document.

Exception handling:

If the maximum amount of comments (currently 100 per document) has been reached, user is informed and Add Comment command is canceled.

Browse Comments

Purpose: browse comments attached to document's signal data.

User input:

- Browse Comments button (toolbar) or Info|Browse Comments... (menu)
- In the Comment dialog
  - comment text edit control
  - Next and Previous buttons to change to the next or the previous comment
  - Remove button to remove current comment
  - Ok, Cancel and Help buttons

Description:

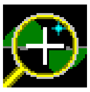
This command is disabled if no comments are attached to the document. If the comment texts are edited, the changes can be discarded with the Cancel button. For further information, see the Add Comment function.

Outputs:

The comment box appears and the first comment is displayed.

Exception handling:

If all comments are removed, "(no comments left)" text is displayed in the disabled text edit control.

 **Zoom In**

Purpose: show a smaller portion of the signal data with greater accuracy on the time axis.

User input:

- Zoom In button (toolbar ) or View|Zoom In command (menu)

Description:

Zooming is one of the most essential functions when analyzing signals visually. The more there are signal data on the display along the time axis, the weaker is the time resolution. Zooming is applied either at the beginning or at the end of each view, depending on the selection in the Preferences dialog. Automatic zooming can be selected and defined to be N milliseconds, where N is typically 3000...10000. When enabled, the preferred time resolution is set automatically after new signals are opened or recorded.

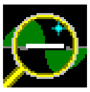
As the signal is zoomed in several times, the zooming factor and position for each zoom level is saved. The Zoom Out commands uses this data to restore preceding zoom levels just as they were. When there are more than one pixels per sample on the time axis, the exact sample positions are shown as small circles in amplitude type views, if that option is selected in the Display Setup | Appearance.

Outputs:

The selected area or, if no selection is currently done, half (along the time axis) of the current view is displayed.

Exception handling:

When the maximum zoom resolution or zoom level has been reached, the Zoom In function is disabled.



Zoom Out

Purpose: show a larger portion of the signal data less accurately on the time axis.

User input:

• Zoom Out button (toolbar ) or View|Zoom Out command (menu)

Description:

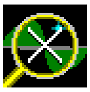
This function is the reverse operation for the Zoom In function. The zooming factor and positions of the preceding zoom levels are restored.

Outputs:

The signal portion of the preceding zoom level is displayed.

Exception handling:

When the signal is zoomed fully out, the Zoom Out function is disabled.



No Zoom

Purpose: zoom fully out, i.e. display the whole signal.

User input:

• No Zoom button (toolbar ) or View|No Zoom command (menu)

Description:

This function has the same behavior as if the Zoom Out command would be repeated until the whole signal is displayed.

Outputs:

The signal is displayed from the beginning to the end. Zoom Out and No Zoom commands are disabled.

Exception handling:

When there are no signal data to be displayed, all zoom functions are disabled.

Load Display Setup

Purpose: load a previously saved display setup from disk.

User input:

- View|Load Display Setup... command (menu)

Description:

Normally, once a preferred and functioning display setup is found, it remains more or less the same. This is not only due to the somewhat fixed hardware setup and capabilities, but also to minimize the complexity of using the software.
Display Setups can be created and saved in the Display Setup dialog. Setup with a name “Default.Dis” is automatically selected for new documents if so selected in the Preferences dialog.

Outputs:

The views defined in the selected display setup appear on the display.

Exception handling:

Views associated with nonexisting channels are discarded.

Turn Monitoring On/Off

Purpose: turn all signal input/output monitoring on or off.

User input:

- View|Turn Monitoring On/Off command (menu)

Description:

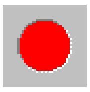
Computer's soundcard is a hardware resource that can't typically be shared. When monitoring is turned off, sound input device is free to be used by other applications.

Outputs:

When a document is not open, the view graphs visualize the input signal if monitoring is turned on. If the scrolling view -option is selected in the Preferences dialog, the most recent input is seen at the right margin of the window; otherwise the drawing position is reset to left margin when right margin is reached. If the level monitoring -option is selected, input signal levels are displayed on the left side and output levels on the right side of the intensity scale bar. Each channel has its own level indicator bar. Levels are displayed in decibels (dB), 0 dB being the maximum amplitude level for the digital signal. Color bar labels are given in decibels, unless all spectrogram views use a non-logarithmic intensity representation. If the focus sensitive -option is selected, monitoring is automatically turned off when the input focus changes to another application window. Processing load percentage is shown on the status bar at the bottom of the window when monitoring is turned on. This percentage shows how much of the available calculation time the real-time signal processing consumes.

Exception handling:

If the processing load exceeds constantly 100%, monitoring is turned off (as well as signal pre-processing), and user is advised to reduce either the sampling frequency or signal pre-processing demands.



Recording

Purpose: save input signal from the soundcard in computer’s memory.

User input:

- Record button (toolbar) or Spacebar (keyboard) or Record command (menu)
- Stop button (toolbar) or Esc (keyboard) or Stop command (menu)
  (needed only with the continuous recording method)

Description:

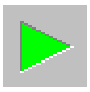
There are two recording methods available. In the continuous mode, recording continues until explicitly stopped by user. In the snapshot mode (default), the currently displayed time period of the signal is recorded.

Outputs:

With the continuous recording method, the record cursor () is shown and the record button appears highlighted during recording. When continuous recording is stopped, normal appearances of the record button and the cursor are restored. Recorded signal is saved in a temporary file on the harddisk and displayed on the screen.

Exception handling:

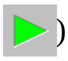
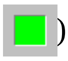
If there’s not enough space on the harddisk for the recorded data, recording is stopped (continuous mode) or discarded (snapshot mode), and an error message is displayed.



Playing

Purpose: play the signal in the computer's memory through the soundcard.

User input:

- Play button (toolbar ) or Enter (keyboard) or Play command (menu)
- Stop button (toolbar ) or Esc (keyboard) or Stop command (menu)

Description:

Current selection or, if nothing is selected, currently visible portion of the signal is played. Play Loop repeats the play operation continuously. Play All plays the whole signal, regardless of what's currently visible or selected.

Outputs:

The play cursor () is shown and the play button is highlighted during playback. A progress indicator is drawn for all views that have an output channel defined. If no output channels are defined, the play button is disabled.

Exception handling:

If output device can't be initialized, user is informed and the play operation is canceled.

Signal pre/postprocessing

Purpose: provide tools for signal processing.

User input:

- Action|Preprocess... or Action|Postprocess... command (menu)

- In the Process dialog:
  - channel list (one channel / row), where each channel has
    - delay setting; delays are given in milliseconds
    - filter selection; available filter definitions are listed in a drop-down list
    - normalization selection (“Yes” or “No”)
  - “Copy to all channels” checkbox
  - Clear button
  - Load... button
  - Save... button
  - Apply button (Preprocessing only) = apply new definitions
  - Ok button = accept definitions and close the dialog
  - Cancel button = discard all changes and close the dialog
  - Help button = display help on using the Process dialog

Description:

Both pre- and postprocessing is provided. During monitoring and recording, signal preprocessing is applied to the input signal in real-time. Postprocessing can be done for saved signals. Apply button applies the selected processing immediately without closing the dialog. When the “Copy to all channels” checkbox is checked, a change in one channel’s processing setup is copied to all other channels. All definitions can be cleared with the Clear button. Definitions can also be saved to and loaded from the harddisk.

Outputs:

The effect of the pre-processing can be seen in the monitor views; i.e. the filtered frequencies can be seen in a spectrogram view. When postprocessing is applied, a position indicator shows the operation progress.

Exception handling:

When requested preprocessing takes too much time to be calculated in real-time, user is informed and preprocessing is turned off.

Preferences

Purpose: setup general options according to the preferences of the user.

User input:

- Setup|Preferences... command (menu) = open the Preferences dialog
- In the Preferences dialog:
  - recording method selection and time window selection for the snapshot method
  - monitor selection checkboxes
  - display time unit selection radiobuttons and automatic zooming selection
  - option to reset display and hardware to default settings for new documents
  - option to display the Diagnosis Information dialog automatically
  - Ok, Cancel and Help buttons

Description:

For information on the two recording modes, see Recording.

For information on the level monitoring, see Monitoring.

For information on the automatic zooming, see Zoom In.

As the display and hardware setup is affected by each opened document, an option is provided for resetting them to default values for all new documents.

By default, the Diagnosis Information dialog appears automatically each time when a newly recorded signal is about to be saved, in order to automate the documentation process. If this selection is turned off, the Diagnosis Information dialog must be always explicitly opened to enter patient and diagnosis information.

Outputs:

In the snapshot recording mode, a time ruler is displayed also during monitoring to indicate the selected time window length. Time is shown in bytes, samples or seconds as selected. When input/output signal levels are monitored, each channel has its own level indicator bar on the left side of the window.

If automatic zooming is requested, each newly opened or recorded signal is zoomed to the specified milliseconds.

Exception handling:
The time variables entered by user (snapshot recording window and the automatic zoom length) are checked and limited by the software to a meaningful range, in which the behavior of the related functions is tested to be correct.

Hardware Setup

Purpose: select the sampling parameters and the i/o hardware to be used.

User input:

- Setup|Hardware... command (menu) to open the Hardware Setup dialog
- In the Hardware Setup dialog:
  - input and output device selection listboxes
  - sampling frequency selection listbox
  - sampling resolution selection radiobuttons (8, 12 or 16 bits)
  - Ok, Cancel and Help buttons

Description:
By default, the input and output hardware is selected automatically by Window. User may override this automation by explicitly selecting the signal input and output devices. These may be features of a single device or two separate devices.
When selecting the sampling frequency, it should be taken into consideration that according to the Nyquist sampling theorem [7], the sampling frequency should be selected to be at least twice as high as the highest frequency of interest in the signal.

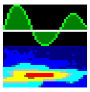
The dynamical range of a digital signal depends on the sampling resolution; with 16 bit resolution, the sample values (describing the amplitude intensity levels of the signal) are in range [-32786, +32767], and the theoretical maximum for the s/n ratio is 20*log10(65536) ≈ 96 dB. With 8 bit resolution, the s/n ratio can be theoretically 48 dB, and much less in practice. 16 bit resolution is recommended to avoid noisy recordings.

Outputs:

Under the output device selection, available sampling resolution and frequency ranges are shown along with the max. number of output channels. These channels can be utilized in the Display Setup dialog, when selecting the output channel for each view. Also, user is let to know whether or not simultaneous input and output (i.e. full duplex) is possible with the selected hardware configuration.

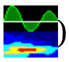
Exception handling:

The software tests the selected devices by trying to open and initialize them. User is informed if any problems are detected.

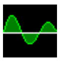
 Display Setup

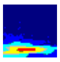
Purpose: setup the display views.

User input:

- Display Setup button (toolbar ) or Setup|Display... command (menu)
- In the Display Setup dialog:
  - Clear button
  - Load Setup... button
  - Save Setup... button
  - Ok, Cancel and Help buttons

- list of views (one view / row), where each view has
  - view index listbox
  - input and output channel selection
  - description text
  - type selection (Spectrogram, Phono, ECG or Respiration view)
  - height selection (automatic or percentage of the available vertical space)
  - display filter selection
  - normalization label (“Yes” or “No”)
  - appearance label = open the Appearance dialog for selected view type:

     Amplitude graph appearance dialog:
    - appearance selections (filling, absoluting etc., see also Zoom In)
    - calculation algorithm selection (amplitude or averaged envelope)
    - amplification coefficient for averaged envelope calculation
    - color selection for the curve envelope and interior
    - sliders for red, green and blue components for currently edited color
    - Restore button for restoring the original colors
    - Defaults button for resetting the colors to default values
    - Ok, Cancel and Help buttons

     Spectrogram appearance dialog:
    - spectrogram type selection (logarithmic, power or linear intensity)
    - FFT window function selection (Hanning, Hamming or Blackman)
    - FFT window size and data size selection
    - frequency band selection
    - linear or logarithmic frequency axis selection
    - musical stave display selection (separately for G and F keys)
    - frequency tracking selection and setup (threshold and frequency band)
    - color palette and palette intensity selection
    - Ok, Cancel and Help buttons

Description:

There is display setup information attached to each .AMR document. This information specifies to number of displayed signal views (graphs) and their properties. The views share the same time axis, and are positioned on top of each other. Each view has an input channel associated with it. If no input channel is selected (indicated with a dash, "-"), the view is removed. Number of available input channels is determined by the soundcard capabilities when a document isn't open. Order of the defined views can be changed by re-selecting view numbers in the leftmost listbox.

Each view can also have an output channel for the playback. Number of available output channels is limited by the capabilities of the soundcard that is selected as the output device in the Hardware Setup. The "mono" setting means all available output channels.

View description is optional, but recommended to clarify the data contents.

Display filtering means that the signal is filtered only for the display; the physical contents remain unchanged. To filter signals physically, use the Processing dialog.

When normalization is selected ("Yes"), the amplitude type views are stretched vertically to use all available space in the view. The actual amount of dynamical range used is shown in brackets on the vertical ruler, i.e. "(75%)". Thus, no stretching can be done if the signal uses all of the available dynamical range. Normalized spectrograms are adjusted to optimal intensity level, also during the input monitoring.

User defined display setups can be saved and loaded. The setup saved in "Default.Dis" is automatically selected for new documents if this option is selected in the Preferences dialog. All view definitions (except the first one - there must always be at least one view) can be cleared by pressing the Clear button.

Outputs:

Effects of the changes on the views are seen when the Display Setup dialog is closed with the Ok button. Changes to the color palette are previewed while the dialog is still open.

Exception handling:

Numerical input values are checked and limited to the valid range. FFT window size is automatically adjusted to be a power of 2. When the maximum number of views is reached, new view definitions can't be made. If all available vertical space is used by the non-automatic heights (expressed as percentages), the sum equals 100% and a new view is created, it is assured that at least 10% of the available space is shared with the automatic height views by decreasing the largest defined height percentage.

Color Setup

Purpose: setup the display colors.

User input:

- Setup|Colors... command (menu) to open the Color Setup dialog
- In the Color Setup dialog:
    - edited color -selection radiobuttons
    - sliders for red, green and blue components for currently selected color
    - Restore All button for restoring the original colors
    - Defaults button for resetting the colors to default values
    - Ok, Cancel and Help buttons

Description:

User interface colors (excluding the view colors defined in the Display Setup|Appearance) can be adjusted here. By clicking a color box on the left side of the dialog, the standard Windows color selection dialog is displayed. Another way to edit the colors is to use the sliders for adjusting the red, green and blue primary components of the currently selected color. Original color is shown on the left half and the changed color on the right half of the color box above the sliders. Original values can be restored with Restore All -button. Colors can be reset to their default values with the Defaults button.

Outputs:

The changes are previewed while the dialog is still open. If the dialog is closed with the Cancel button, original colors are restored.

Exception handling:

When other color intensive Windows applications are running simultaneously, and display is in 256 color mode, all colors may not be displayed correctly due to the limited system resources.

Filter Setup

Purpose: create and edit filter definitions.

User input:

- Setup|Filters... command (menu) to open the Filter Setup dialog
- In the Filter Setup dialog:
    - Ok, Cancel and Help buttons
    - list of filters (one filter / row), where each filter definition has
        - filter index listbox
        - filter name text
        - type selection (Butterworth, Chebyshev or Special)
        - band selection (lowpass, highpass, bandpass or bandstop)
        - cutoff frequency
        - bandwidth
        - filter order
        - Information column for opening the Filter Information dialog

Description:

Filters are used both for signal pre/post-processing (see Process dialog) and signal graphs (see Display Setup). A new filter is defined by giving it a name, after which the other fields are initialized with default values. The sampling frequency dependent coefficients are calculated automatically for Butterworth and Chebyshev type IIR filters. In addition there's the Special filter type to provide the user the possibility to give the filter coefficients manually in the Filter Information dialog. For Special type filters, it is recommended to use the intended sampling frequency as a part of the filter's name.

Band selections have the following meaning:

- lowpass: filter all frequencies above the given cutoff frequency
- highpass: filter all frequencies below the given cutoff frequency
- bandpass: filter all frequencies outside the band [cutoff, cutoff + bandwidth]
- bandstop: filter all frequencies inside the band [cutoff, cutoff + bandwidth]

As the filter order increases, more calculation time is required.

In the Filter Information dialog, the filter coefficients are shown (and given for Special type filters). Also, possible errors detected in the filter design and calculation process are displayed (for other than Special type filters).

Outputs:

The defined filters can be used both in the Process dialog and in the Display Setup dialog. If a filter definition is currently in use in either of these dialogs, the changes take place as the dialog is closed with the Ok button.

Exception handling:

All input values are checked and limited automatically. If overflows or any errors are detected during the filter design, warning messages are displayed in the Filter Info dialog.

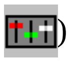
Tool Applications

Purpose: provide a quick way to launch other applications with user defined parameters.

User input:

- Tools|<user selected name> command (menu) to launch the selected application
- Tools|Tool Applications... command (menu) to open the Tool Applications dialog
- In the Tool Applications dialog:
  - Ok, Cancel and Help buttons
  - list of tools (one tool / row), where each tool definition has
    - tool index listbox
    - description text (to be shown in the Tool menu)
    - application selection
    - command line parameters

Description:

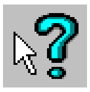
Tool application is any other program that the user wants to have quickly available.
In the Tool Applications dialog, these applications can be selected, along with optional command line parameters. Also the name of the currently open document can be given as a command line parameter (represented with a %-sign). Each tool is displayed in the Tool menu with the description given by the user. Mixer tool (toolbar ) has been given as an example of a tool application.

Outputs:

Tool definitions are updated in the Tool menu when dialog is closed with the Ok button.

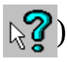
Exception handling:

If the tool application has been moved or deleted since the definition, and can’t thus be found when selected in the menu for launching, user is informed and a possibility for locating the tool application in question is offered.

 **Help system**

Purpose: provide help on using the program and its functions.

User input:

- Help|Contents... command (menu) to open the Help window contents page
- Help|Using Help... command (menu) to open instructions for using the Help system
- Context Sensitive Help button (toolbar ) to activate the context sensitive help

Description:

On-line help system can be used either by browsing the help pages via the Help|Contents command, or by using the context sensitive help. This is done by

- clicking the Context Sensitive Help button, activating the help mode
- selecting a menu command or clicking any button (or other object) on the screen
- the help window appears, providing help for the selected object or menu command

Context Sensitive Help mode is deactivated automatically after the help window appears, but it can also be deactivated by pressing the Escape button.

Outputs:

As the Context Sensitive Help mode is activated, the cursor shape is changed to resemble the Help button. When context sensitive help is deactivated, the cursor shape is restored. Help pages are displayed in a separate window.

Exception handling:

If help can't be found for the selected item, user is informed by the Windows help system.

Mouse functions

Purpose: make selections or select a position in the signal graphs.

User inputs:

- left button click on the views = set current position
- left button click with the keyboard Shift key down
  = select area from the current position to the click position
- right button click to open a floating pop-up-menu
- mouse movement while the left mouse button held down =
  - comment mark dragging if the button is pressed down on a comment mark
  - otherwise, signal area selection (if there are signal data currently displayed)

Description:

By default, the position indicator is set to the beginning of the signal. Current position is indicated with a vertical line. Position is used, for instance, by the Play function as a starting position for the playback and in the Paste functions as a starting position for the pasted data. Selections are used by several functions, such as Play, Cut, Copy, Delete, Trim and Zoom In. Some of the most essential commands, such as Play and Add Comment, are found in the floating pop-up-menu. Add Comment Here! is the only command that is available only through the floating pop-up-menu.

Outputs:

When a new position is selected, the position indicator is moved to the selected position. Selected area is indicated with a dot-matrix between two vertical lines. Selection disappears when a new position is selected by clicking the left mouse button.

Exception handling:

If the size of the signal data changes, is it ensured that position and selection are within the valid range.

4. Discussion and Conclusions

When analyzing a project, typically a good question to ask is whether or not the results meet the specified requirements. Initial requirements for this project were rather loose, since it wasn't known what exactly would be the goal. Specifications became more detailed as satisfying implementations were found. Some specifications had to be removed and others were added as the realizability and usability of ideas was tested in practice. An example of discarded ideas was a "dictation machine" as an alternative way to fill in the patient information. At first it sounded like a promising idea as something that cardiologists are already familiar with; however, several technical and usability problems showed that it would probably just cause everybody a bunch of gray hairs in the future. On the other hand, the snapshot recording mode is a feature which improves the usability of this software significantly, but was invented only after a couple of years of development. It was received by the test personnel with great enthusiasm, and has now established it's place at the heart of the program's functionality.

"Shoot anything and call it the target" is one of the ways to guarantee success. In this project, the definition of the goal has been, to some degree, a subject to change. Features and objectives for the product are defined in the requirements specification, chapter 2. These have been essentially fulfilled, and even exceeded: with only minor changes, the software can be made suitable for monitoring and analyzing practically any biosignals. Still, the ultimate goal must be set much further.

The existing product remarkably improves the objectivity and accuracy of diagnosing heart murmurs. Integrated multimedia documentation increases comparability and provides means for post examination and educational use. Electrically exchangeable documents increase the time and cost efficiency of diagnostic work by making it possible to examine patients already in the primary health care and utilize teleconsultation when necessary. However, only a few people are using it in their work on everyday basis so far. Thus, the final success still remains to be seen.

References

[1] Sjöholm Johan, *"Development of a Phonocardiography Amplifier*",
Master's Thesis, spring 1999, Helsinki University of Technology

[2] Moss and Adams, *"Heart diseases in infants, children, and adolescents"*,
Williams & Williams, Baltimore, 1995, pp. 135-146

[3] A. Van Oort, M. Le Blanc-Botden, T. De Boo, T. Van Der Werf et al.,
"The vibratory innocent heart murmur in schoolchildren: Difference in auscultatory findings between school medical officers and a pediatric cardiologist",
Pediatr. Cardiol., 1994, vol. 15, pp. 282-287

[4] Lehner R.J. and Rangayyan R.M., *"Phonocardiogram signal analysis: A review"*,
CRC Critical Reviews in Biomedical Engineering, 1988, vol. 15, pp. 211-236

[5] Sommerville Ian, *"Software Engineering"*, 5. ed., Addison-Wesley, Harlow, 1995,
pp. 7-12

[6] Jackson L.B., *"Digital Filters and Signal Processing"*,
Kluwer Academic Publishers, Boston, 1986, pp. 153-167

[7] Rossing Thomas D., "The Science of Sound", 2. ed., Addison-Wesley, Reading,
1990, pp. 556-558