Remote Biosignal Monitoring, Display, Online Analysis and Retrieval

M Micalizzi, F Conforti, A Macerata, C Passino, M Varanini, M Emdin

CNR Institute of Clinical Physiology, Pisa, Italy

Abstract

A model for distributed signal acquisition and processing has been designed and implemented. The model is based on a LAN client-server structure: it allows signal acquisition on a dedicated server and acquired samples distribution to different client workstations for signal representation and/or processing.

The communication between server and clients is based on TCP/IP protocol. A set of specific commands allows the client to get the server acquisition configuration and to request/receive signal samples. In order to facilitate the development of client own system for signal analysis a library of JAVA functions (or classes) was created including the basis of biosignal processing and display; these classes have to be included by the developer into the user specific program for signal acquisition and elaboration.

The system is under test in the research LAN of our Institute. It consists of one PC-server with an A/D converter board and one client workstation for displaying and processing cardiovascular and respiratory signals obtained from patients studied with autonomic testing.

1. Introduction

Acquisition and analysis of biological signals (such as ECG, arterial pressure, respiration, etc.) is a common procedure in clinical departments and medical research laboratories: signals from the same patient can be analyzed by different processing procedures to extract diagnostic information and should be represented in different display formats to enhance their specific features.

Commercial instrumentation does not generally allow sharing of data between applications, forcing the analysis to multiple time shifted sessions. Moreover, in absence of an agreement for standard formats of data among biomedical companies, different commercial devices should be employed, each devoted to a specific task and including its own acquisition module and internal data storing pattern.

To overcome this wasting of resources a model of a new structure for signal acquisition and distribution is proposed. This structure keeps separate the two main processes: acquisition and processing.

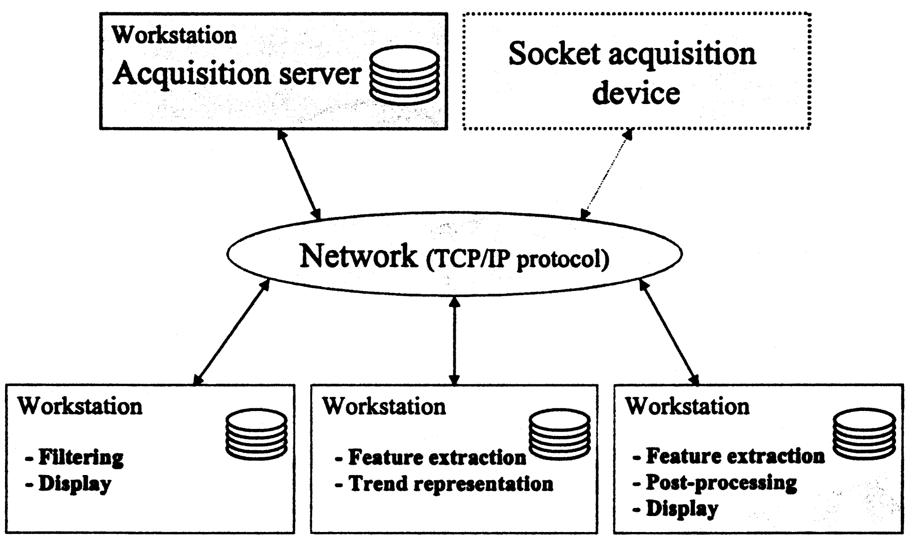

Figure 1. System structure

This results in optimization of resources, sharing of data among different workstations and applications and in increase of flexibility in terms of number and kind of computers involved in the overall analysis process applied to the same signals set.

2. System structure

The acquisition-processing model is based on a client-server structure (Fig. 1) distributed on a Local Area Network (LAN). The server subsystem is devoted to the acquisition of data from external sources, meanwhile client subsystems provide specific tasks on signal data according to the user requirements. Server and client communicate through the network, realizing a distributed system where the same input data can be processed on different workstations.

2.1. Acquisition server subsystem

The server is designed to manage data acquisition and distribution (Fig. 2). It performs four basic actions: input configuration (type of source, number of channels, gain, offset, polarity, etc), getting data from input sources, saving signals samples in a local temporary memory, making the data available to any client.

The server is able to acquire data from different sources: A/D boards, serial port, USB port and disk file. Each type of source is managed by specific software drivers which are developed as DLL. The main program of the server uses a common user-interface for all the input sources and for the acquisition setting. In each acquisition session, the server has to perform always the following steps: configuration, start, checking (acquisition loop), cleaning up.

The protocol of each acquisition session can be stored on local disk, saving all the input features and configuration, for next easy recall and fast restart.

The server allows event recording, such as text note or time marker, linked to each acquisition session for detailed history of all the procedures adopted and occurring events.

All the input signals can be normalized by a calibration functionality which can be recalled and applied at any time during the acquisition; it will be active for the whole period between different calibration time points.

Moreover, the server can be used as signals generator, playing file data in cyclic mode or generating synthetic multicomponents signal waves at different frequencies and amplitudes.

The server software was developed in C++ language and it runs in Windows environment.

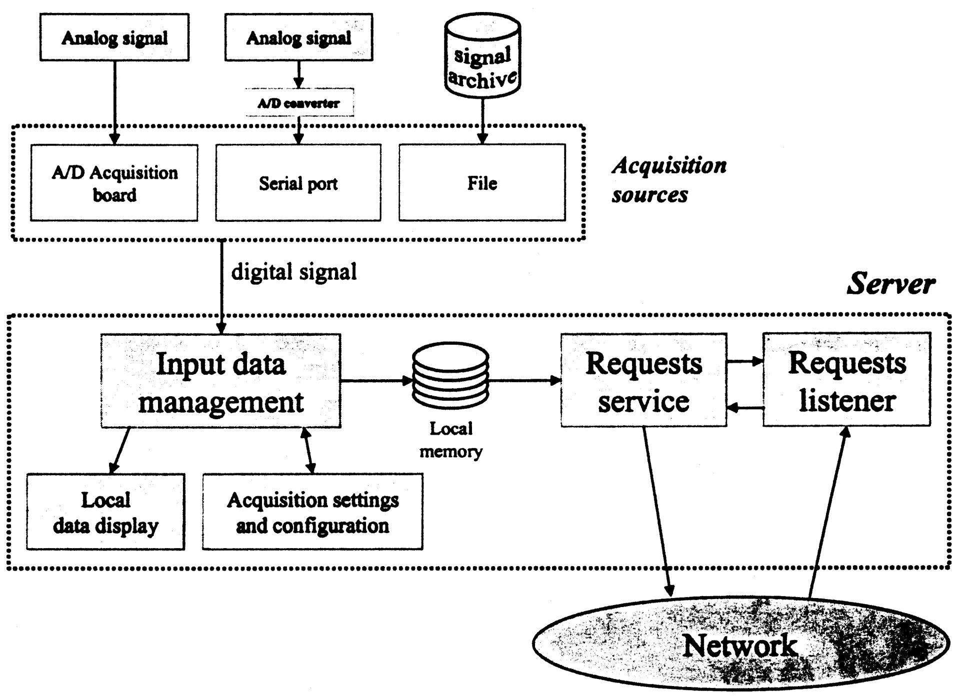

Figure 2. Server structure

2.2. Client subsystems

As client subsystems are defined all the workstations distributed in the LAN hosting application programs linked to the server through TCP/IP protocol [2]. The client (Fig. 3), synchronized with the server and recovering data through the network, is devoted to the manipulation of data such as signal display and processing, data retrieval according to the user needs.

The client communicates with the server by using sockets: sockets are the basis of the TCP/IP communication protocol, commonly used by computers for connecting in a local area network or in Internet. Sending/receiving information through this protocol does not depend by the programming language or operative system [3].

We have defined a specific common language used by both client and server to manage communications among them; this language consists on a set of commands to get source status information and to allow the exchange of data between the acquisition-server and processing-clients. The exchange of information follows a procedure similar to a Telnet session: a client can be directly connected to the server by Telnet asking for acquisition setting configuration or getting data by using predefined commands and proper parameters.

This procedure offers a great versatility and portability, being supported by any hardware and software platform.

A library of Java routines was developed to promote the system flexibility. Java classes cover the basic functions of signal processing and display, including different formats of signal representation (single/multiple traces, X-Y view, overlapped data, trend, etc.), windows management, filtering, ECG and hemodynamic pressures analysis with characteristic features extraction. These functions can be included as Java classes in new applications lightening the development of dedicated signal processing and display programs.

The power of this client-server structure is that the information is shared through the network; there is no need of using the same computer for the acquisition and the processing. The user (client) may choose the most suitable platform depending on his needs, taking advantage of the machine speed or video graphic capability or other characteristics of the host computer.

For example, a basic PC can be used for signals acquisition while a very powerful workstation is performing complex calculations, without affecting the delicate process of acquisition on the server. This results in an optimum use of available resources; several clients, configured with the same application program, can be connected to the same server, dealing with the same data realizing a multidisplay real time monitoring in the same patient. Furthermore, more servers can be connected to different patients and a single client can collect and display all the patients data obtaining a multi-bed monitoring unit.

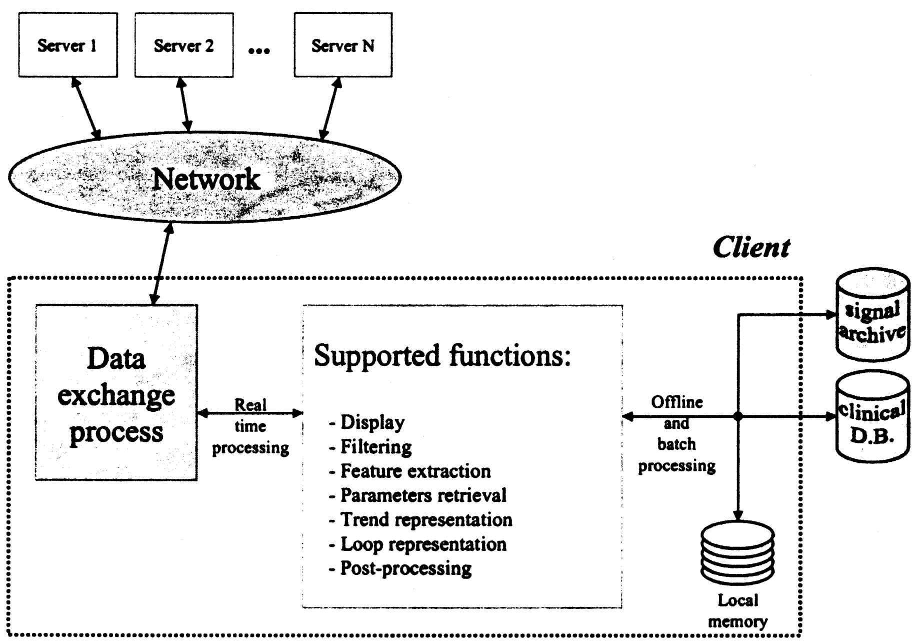

Figure 3. Client structure

3. A system for pathophysiological research

In the medical departments of our Institution, many patients are submitted to multiparametric recording in baseline conditions as well as during physical and/or pharmacological stress testing, in order to evaluate their status and eventual impairment in regulatory control of cardiovascular system. A prototype of the acquisition-processing model was implemented as a system for the Physio-Lab (laboratory for neuroendocrine studies) in our Institute.

The server module manages an A/D board and a 12 channel electrocardiograph connected through the serial port, allowing acquisition of different signals such as ECG, noninvasive arterial pressure (Colin Medical Instruments) and respiration.

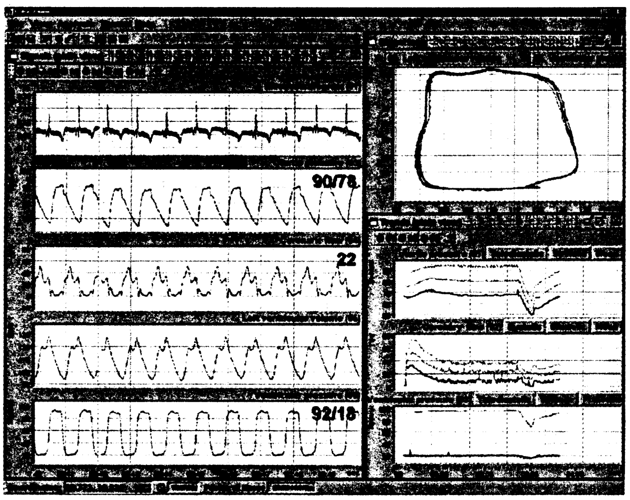

Figure 4. Client layout display of signals (left), parameter trends and loop view (right).

The client consists of a PC hosting a Java program developed for exhaustive analysis and checking of stress tests.

In the client program, a set of digital filters can be applied to clean up the input signals or the features trends. The user can select the number and kind of features to be extracted from the signals, to be saved in a file and to be displayed by different formats of graphic representation, from standard time-course to X-Y view (Fig. 4). Utilities are available for forward-backward time scrolling of the signals and for in/out zooming.

The client software program allows a personal configuration of the screen through several instances of windows, just like in a multi-document application.

The client can be configured for local display and processing, having as input a file with previously acquired, and eventually processed, data without any

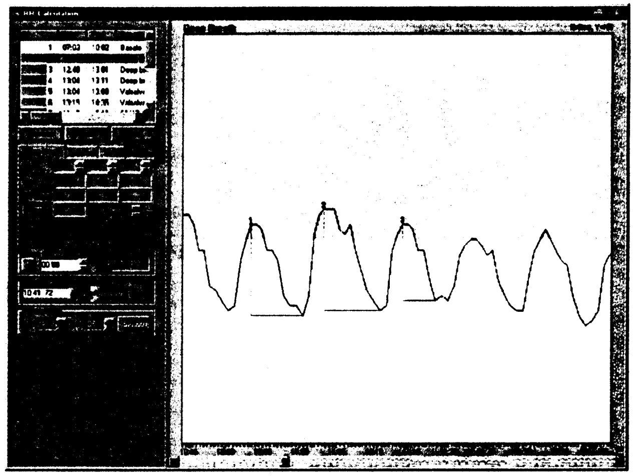

Figure 5. Example of post-processing: RR time series computing respiratory sinus arrhythmia during deep breathing autonomic testing.

connection to the server. Thus, the user may review the results of previous analysis and process again the data by extracting new features or submit them to new algorithms (Fig. 5). The results of this post-processing can be saved into a new file or can be used to update the original file.

4. Conclusions

A model for distributed acquisition and processing is proposed, meeting the needs of signal multiprocessing and resource optimization. The system is characterized by a great flexibility and portability.

The separation of acquisition and processing function allows to independently cope with the needs of acquisition and processing, both for the hardware and the software. Moreover, this methodology allows an easy upgrade of the overall system. For instance, adding a new A/D board or changing completely the platform of the server, does not affect client functionality.

On the other hand, client hardware/software can be updated, or new client workstations can be developed and installed within the network, without any care of the acquisition process.

References

[1] National Instruments Corporation, NI-DAQ® Software Reference Manual, 1999

[2] TCP/IP Network Administration, Craig Hunt, 1998

[3] Bob Quinn, Dave Shute. Windows Sockets Network Programming. Addison-Wesley, 1995.

Mauro Micalizzi, Ph.D.
CNR Institute of Clinical Physiology
Via Giuseppe Moruzzi, 1, 56124 Pisa, Italy
mica@ifc.pi.cnr.it