INTEGRATION SOLUTION FOR THE ACCESS TO HETEROGENEOUS MEDICAL DEVICES
Communication with Healthcare Devices in Intensive Care Units

Susana Martin-Toral, Jose Luis Rodriguez-Gonzalez
Computer and Information Technologies Division. CARTIF Foundation
(Centre for Automation, Robotics and Information and Manufacturing Technology)
Ave. Francisco Valles 204-205, Parque Tecnologico de Boecillo, Valladolid, Spain

Javier Perez-Turiel
Head of Biomedical Engineering Division, CARTIF Foundation

Keywords: Medical devices, drivers, VITAL (Vital Signs Information Representation), manager/agent architecture, MDIB (Medical Data Information Base).

Abstract: This paper presents a free Critical Care Information System (CCIS) that shows an essential infrastructure for critical care medical and nursing practice. Specifically, a Patient Integral Analysis Aid System (SAIP) in Intensive Care Units (ICU) has been developed to cover the needs discovered in these scenarios. An importart part of this system is related to medical equipment, that offers important information to help in medical diagnosis. ICU patients are usually connected to several of these devices which register their physiological parameters. The integration of these devices, in order to exchange the generated information, is difficult because they are developed by different manufacturers and with different communication protocols and information representations. Due to this, it has been necessary to develop a set of communication drivers for each medical device, according to the current regulations. To reach this objective, the developed drivers have a common interface for the access and collection of medical device data. The main goal of the present paper is to show the work done to obtain a real interoperability among medical devices from different manufacturers and with different communication protocols in ICU services for automatic data collection, storage and retrieval.

1 INTRODUCTION

In this paper a free Critical Care Information System (CCIS) is presented. CCIS is an essential infrastructure for critical care medical and nursing practice. This premise is described in terms of the critical care environment, clinical decision making in critical care, increasing demands for information about quality and costs, and national initiatives for the sharing of healthcare information. This kind of systems are designed to collect, store, organize, retrieve, and manipulate all the data related to the care of the patient in Intensive Care Units (ICU). Some of the main purposes of these systems are: automated vital signs capture, cardiac rhythm analysis and dysrhythmia detection, reporting laboratory results, entry and transmission of physician orders, admission, discharge and transfer of patients, calculation of medication doses, fluid infusion rates, shift, and daily intake and output volumes, organization of patient records, calculation of patient plan of care, entry and organization of care documentation and prompting the caregiver for recorded documentation (Adhikari and Lapinsky, 2003; Fraenkel et al., 2003; Frize et al., 1997). In this research project a Patient Integral Analysis Aid System (SAIP) in Intensive Care Units (ICU) has been developed, which is now being used in the ICU of the University Clinical Hospital of Valladolid, in Spain.

The SAIP is a system based on free software GNU GPL (General Public License) (Ope, 2007) composed by a set of computer applications to facilitate and improve the attention received by the patient in intensive care units (Martin et al., 2004; Jose, 2007). It provides a support system to collect, store and manage the clinical information in the ICU, with the purpose of facilitating the daily medical routine and to improve the quality of the attention provided to the patient by overcoming the actual difficulties of inter-

Martin-Toral S., Luis Rodriguez-Gonzalez J. and Perez-Turiel J. (2008).
INTEGRATION SOLUTION FOR THE ACCESS TO HETEROGENEOUS MEDICAL DEVICES - Communication with Healthcare Devices in Intensive Care Units.
In *Proceedings of the First International Conference on Health Informatics*, pages 247-253

connection and storage (Shortliffe, 2001).

The ICU patient is, habitually, connected to one or several devices - called in this paper medical devices - that allow information on several of their physiological parameters to be measured, as shown in figure 1 with a vital signs monitor. These devices generate a great amount of information, but their capacity of storage is limited. In addition, they are usually from different manufacturers and models, and they are not interconnected with the rest of the devices. That is why the integration of the generated information is difficult. This heterogeneity of medical devices is one of the main problems for making an integral system for the ICU. Each device has its own communication protocol and its own connection interface, for example, RS-232, Ethernet, MIB (Medical Information Bus), etc. and there is no interface or protocol that allow the unification of communication and access between different devices and a user application that wants to communicate with them. In order to solve this problem, the developed system unifies, for each type of medical device, the data obtained from the different medical devices that compose the system, by using diverse standards of denomination and storage.

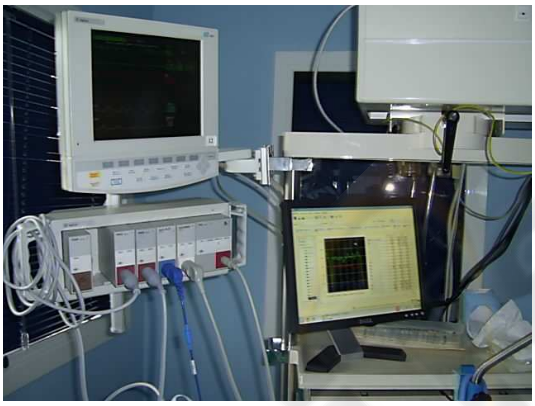


Figure 1: Monitor connected to the ICU room PC.

The initial situation in the ICU consists of a series of medical devices, such as infusion pumps, respiratory ventilators and vital signs monitors, each with a different type of connection, a different communication protocol and without a common interface that allows homogeneous access to the data that these devices provide (Martin et al., 2004; Jose, 2007). Due to this, it has been necessary to develop a set of communication drivers for each medical device that allow the information provided by the medical devices be collected and stored automatically, although these are from different manufacturers, as well as the information contributed by the medical staff. Because of that, the drivers developed for the SAIP system have a common interface for the access and collection of data from the medical devices.

Although the main objective of the present paper is to show the work done to obtain the homogeneous interoperability between medical devices in the ICU, additional objectives of the SAIP system (Jose, 2007) are summarized as follows:

- Automation of the medical workflow: electronic management of documents.
- Collection and storage of other types of manually acquired information: medical devices without possibility of automatic capture of information, connectionless devices and data from the patient exploration task filled in directly by medical staff.
- Calculation of medical statistical indexes: APACHE, NEMS,etc.
- Consultation of registered information. The information stored in the system is readily accessible.
- Document elaboration of patients' management (registries, personal information, etc).
- The SAIP must comply with the National medical information security laws.
- Interchange of data with other hospital services. The intensive care unit is not an isolated service, so it needs constant communication with other hospital services. The efficient implementation of defined interfaces should ensure communications between the diverse systems/services of the hospital.
- To manage a safe and trustworthy access to the system. Most of the information generated in hospitals is personal medical information and therefore is confidential.
- Development under the concept of free software (GPL license). The development of applications based on free software is an area of activity of ever growing interest that generates numerous expectations, given the ample possibilities it offers.

The rest of this paper is organized as follows : section 2 describes the proposed control architecture of the SAIP system in the ICU, paying special attention to the part related to the integration of medical devices. Section 3 describes the communication drivers developed to allow the medical devices to communicate with the SAIP System. In addition, it describes the dynamic libraries of plugin type solution proposed. Section 4 describes the agent entity whose function is to communicate with the different medical devices with its associated database. This database stores the device data. In section 5 the results of the system development are presented. Finally, a summary of the work done and some conclusions are included in section 6.

2 ARCHITECTURE SOLUTION

The SAIP is a system based on the use of computer, control and communication technologies to generate a set of computer applications whose main objective is to ease and improve the attention received by patients in intensive medical services. From a physical point of view, the planned system requires the connection of medical devices (vital signs monitors, ventilators, infusion pumps, etc.) to external elements, the installation of computer systems and communication networks, and the development of software applications for data recording, storage, processing and access.

The proposed system architecture is based on a manager/agent framework, and in general it can be seen as a three layer architecture (data, business logic and presentation). Furthermore, the system functionality has been divided in two different parts: the integration environment and the execution environment or healthcare information system (HIS) (Scherrer and Spahni, 1999; Martin et al., 2004).

In general, the aim of the *integration environment* is to unify the information obtained from every kind of system device (both medical and computer). To achieve this objective, several network device manager frameworks have been used (Leinwand and Fang, 1995), such as the ones based on the OSI (Open System Interconnection) interconnection basic reference model or the ones based on SNMP (Simple Network Management Protocol) (Stallings, 1999; Case et al., 1990). On the other hand, the *execution environment*, or HIS environment, is the one in charge of implementing the business logic, so it develops the thickness functionality provided by the SAIP. The modules that form the HIS environment are located throughout the diverse physical devices that house the application, based on the information type they handle. This information is provided by both the integration environment and users, or through other applications and external systems (admission information, laboratory results, etc.). Therefore, the execution environment has the elements needed to process the information and to provide the functionality required by users.

From this architecture, the present work emphasizes the part dedicated to the *communication and information processing provided by medical devices*, according to norms (CEN/CENELEC, 1993; CEN, 2000). The goal pursued in this aspect is to have a system that allows the interaction with patient vital support devices, so as to make the data and alarms generated by these devices available for users (doctors, nursing staff, nursing assistant, etc.). Since these medical devices are in the patients' emergency rooms, the computer equipment entrusted to interact with them must also be located in the same emergency rooms. This IT equipment (pesonal computer) is called, in our system, *ICU room PC*. The ICU room PC architecture is shown in figure 2, and its main characteristics will be detailed in the rest of the section.

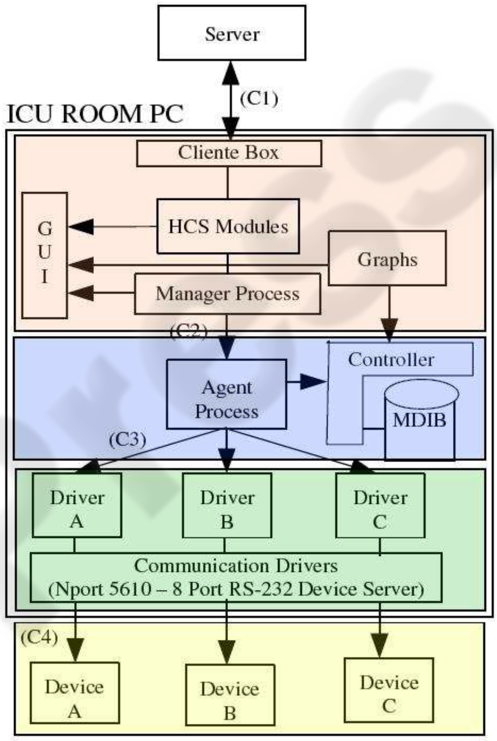


Figure 2: Block diagram of ICU Room PC.

For the ICU room PC connection architecture, the main objective of the integration environment is to unify, for every type of medical device and as far as possible, the data obtained from the different devices that compose the system. This environment, therefore, converts and translates the hardware dependent information provided by medical devices into an annotation and nomenclature understandable by all the devices. Moreover, it also unifies the way in which the calls are made to access information. In this way an integrated and homogeneous medical device management is obtained.

With these objectives, the integration environment uses some parts of the VITAL norm (Vital Signs Information Representation) (CEN, 2002) to define the semantics of the information provided by medical devices as a structured set of objects and methods to represent medical information. The VITAL experimental norm provides the definition of an independent common device representation of vital signs information, and the definition of a common model for accessing to this information. It means that VITAL looks for real interoperability between medical devices even if they are from different manufacturers (CEN, 2002).

In general, the information provided by medical devices is transformed into VITAL nomenclature to be stored in a database called MDIB (Medical Data Information Base). An agent entity is responsible for updating this database, which in addition attends to the requests made by a manager entity, as can be seen in detail in section 4. In short, the integration environment needs:

- A set of drivers that allow communication with diverse medical devices. Each one is device dependent, so they must speak the same language as the device they access.
- The translation of device dependent information into VITAL nomenclature.
- An agent entity that gathers this information periodically.
- A database where information is stored.

In figure 2 the connection schema of the elements housed in the ICU room PC is shown. Three differentiated groups are distinguished in this schema. From the lower level of abstraction to the upper, firstly the set of drivers can be seen, then the agent entity and its related database, and finally the business logic and the graphic user interface housed in the ICU room PC. Each of these groups is presented in detail in the following sections.

3 COMMUNICATION DRIVERS

Drivers are programs in charge of serving as middlewares between the computer operating system and the different elements connected to it. In this particular case, the system works with two different kinds of drivers, as can be seen in figure 2. The first one fits the definition given before, and is called "communication driver" in the figure. This driver allows the communication with medical devices through a communication element. Currently, the system works with a unique communication driver, that is, a serial port hub of Moxa trademark (Moxa's NPort 5610 8 Port RS-232 Device Server). The reason is that almost all the medical devices provide their output through an RS-232 interface, so we have chosen to work with this type of device server, that allows a greater the number of RS-232 interfaces than a conventional computer. Perhaps in the future the system could need other types of drivers for medical device communication, maybe through Ethernet or MIB (Medical Information Bus) interfaces.

The second driver type presented in the schema is related to special programs developed to implement the communication protocol of each medical device. Therefore, there must be a driver for each type and/or model of medical device. These drivers can be seen as translator gateways, since they translate data requests from the agent entity, who makes them in a homogeneous way, into a language understandable by the corresponding medical device, and then they respond in a standard VITAL format. In this way the agent entity can talk to the different devices using the same language, and without worrying about which device it is talking to. The main objectives of these drivers are:

- To implement the communication through the serial port (extendable to other communication interfaces).
- To implement the proprietary communication protocol of each medical device.
- To attend to manager entity requests.
- To translate the information provided by devices into VITAL format.

In this work, a plugin is considered a driver implemented for every type or model of medical device (the second type of driver seen before), since its compilation is made as plugin libraries that can be loaded later by an agent entity. These plugins have been developed using the same interface, so the agent can call their functions in a homogeneous way and without the need to know the device with which it works, or from which manufacturer it is. In this way, the system is totally scalable in order to increase the number of medical devices to work with, and it ensures the communication with heterogeneous devices from diverse models and manufacturers.

When the incorporation of a new medical device is needed in the ICU, and whenever the new device presents some external communication port and protocol, it will be enough to implement the appropriate driver (filling the specified interface and VITAL representation), to generate the plugin library and to inform to the agent entity about the new plugin name and location. Once the plugin is incorporated into the system, the agent entity will be able to communicate and extract information from this device.

4 AGENT ENTITY AND ITS RELATED DATABASE

The agent entity is the element in charge of the communication with the different medical devices (connected to the ICU room PC by the serial ports hub) through the corresponding plugins. Its main target is to collect the information as well as the alarms provided by the different medical devices, to be monitored and validated. Every agent entity communicates with the medical devices assigned to a specific patient and, with the data provided by them, must fill in the database named MDIB, and must respond the requests from a manager entity. Therefore at this level it is a manager/agent architecture.

The main functionalities of the agent entity are:

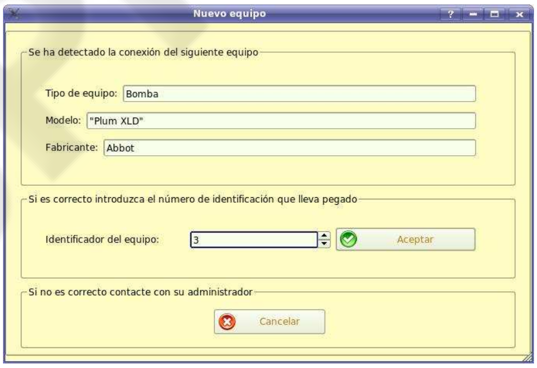
- To detect the connection of a new medical device in the serial port hub assigned to one ICU room and one patient. This detection is carried out by the own hub, which is able to send an SNMP trap to the ICU room PC indicating the status changes in DCD and DSR signals in each hub's port.
- To identify the connected medical device, that is, to detect the type of device the agent entity must work with to use the suitable plugin. With this aim, the agent entity loads every plugin and checks if it exists an appropriated communication with the device related to the plugin. If the device responds its plugin will have been detected, and therefore the agent entity will know the type of device which it works with.
- The previous items provide the system with a plug-and-play functionality related to the connection of medical devices to the system. This functionality simplyfies the connection of these medical devices to the nursing staff.
- To collect periodically the data provided by these devices, and update the MDIB with them.
- To detect alarm conditions in the devices using pull or push methods, according to the behavior of each one.
- To notify the detected alarms to the adecuated element of the systems for its processing and broadcast.
- To attend the requests of the manager entity, that must ask for and provide the information from medical devices to the business logic layer, to be processed and showed to system users.

The MDIB, database containing the information provided by medical devices, has an information model according with a part of the structure and nomenclature of the VITAL norm. This part is used to represent the different information objects. Either the agent entity, or other modules and elements, could need to access to the MDIB database.

5 MAIN RESULTS

In order to understand the obtained results it has to be considered the scenario where the project has been developed:

- Each ICU room has a ICU room PC.
- An ICU room can have more than one patient.
- For each patient there is an agent entity and a MDIB associated to it. Therefore in each ICU room PC there will be so many agent entities and MDIB as patients are in the ICU room.
- All the data gathered during a patient's stay in the ICU is stored in the MDIB.
- This MDIB is filled in a periodical way. In every period the data collected by the drivers is used to fill the database.
- When a patient leaves the service, and therefore the system, the data stored in the MDIB relatec to that patient is erased.



Se ha detectado la conexión del siguiente equipo

Tipo de equipo: Bomba

Modelo: "Plum XLD"

Fabricante: Abbot

Si es correcto introduzca el número de identificación que lleva pegado

Identificador del equipo: 3

Si no es correcto contacte con su administrador

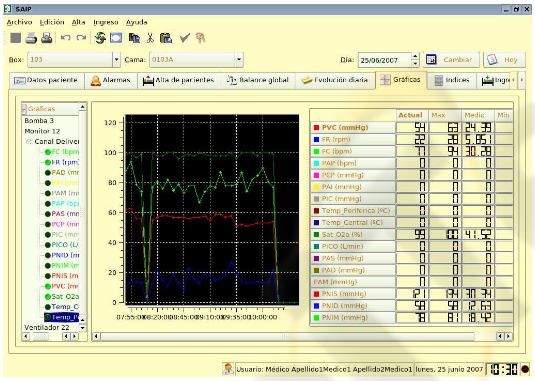
Figure 3: Device identification screen.

- The agent entity must be always active, waiting for the connection/disconnection of a medical device, even if there is no patient in the room.
- When there is no patient in the room and a device is connected, the agent assumes that there is a new patient in the room.
- Whenever a device is connected, if the agent is able to recognize it, a screen appears indicating the device detected (type of device, manufacturer, etc.). If the displayed data is correct, the user must introduce the number that identifies that device univocally, as it is shown in figure 3.

- The agent monitors each device using its sampling period. The data is stored periodically in the MDIB of the ICU room PC, whereas the detected alarms are sent to the appropriate process to be handled. An example of an alarm originated by an infusion pump is shown in the next listing (taken from the standard output):

```
Hilo alarmas: Se ha detectado 1 nueva
              alarma en el equipo 3
----------------------------------------
Identificador de dispositivo: 3
---> Hilo alarmas: ALARMA 0:
        Equipo: Bomba
        Modelo: "Plum XLD"
        Descripcion: Fallo en cassette.
        Tipo de alarma: tecnica
        Prioridad: baja
        Fecha de produccion: 22/06/2007
        Hora de produccion: 08:14:32
----------------------------------------
```

- Once the data is stored in the MDIB, the medical staff has the possibility of visualizing it as graphs (trend curves) corresponding to different physiological signs from the medical devices. Two examples of these trend curves can be seen in figures 4. and 5. Only the vital signs with a green spot are showed in the graph.



	Actual	Max	Medio	Min
PVC (mmHg)	54	63	24.39	
FR (rpm)	22	28	5.851	
FC (bpm)	77	94	30.28	
PAP (bpm)	0	0	0	
PCP (mmHg)	0	0	0	
PAI (mmHg)	0	0	0	
PIC (mmHg)	0	0	0	
Temp_Periferica (ºC)	0	0	0	
Temp_Central (ºC)	0	0	0	
Sat_O2a (%)	99	100	41.52	
PICO (L/min)	0	0	0	
PAS (mmHg)	0	0	0	
PAD (mmHg)	0	0	0	
PAM (mmHg)	0	0	0	
PNIS (mmHg)	121	134	30.34	
PNID (mmHg)	58	58	12.63	
PNIM (mmHg)	78	81	18.42	

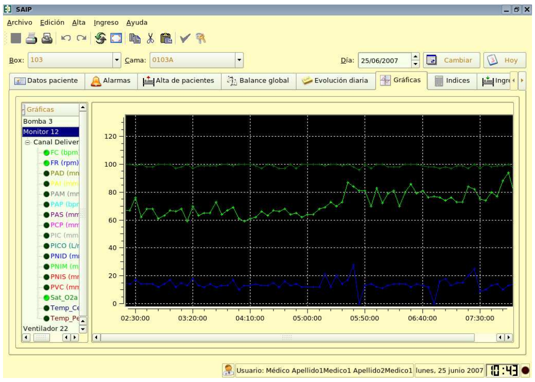
Usuario: Médico Apellido1Medico1 Apellido2Medico1 lunes, 25 junio 2007 10:30

Figure 4: Trend curves from patient monitor data.

- When the patient leaves the service all the medical devices must be disconnected, while the agent entity waits for the arrival of a new patient.

During the project, the following medical devices communication drivers have been developed, tested and included into the system:

- Lifecare XL Infusion Pump from Abbott Laboratories.
- Evita Ventilator from Drager Medical.
- Vital signs monitor from Agilent Technologies.



Usuario: Médico Apellido1Medico1 Apellido2Medico1 lunes, 25 junio 2007 10:43

Figure 5: Detail of trend curves.

The software developed in this project has been implemented using the C++ language and the Trolltech's QT library (Tro, 2007).

6 CONCLUSIONS

In this paper it has been presented the components of the SAIP system developed to interact with medical devices (vital signs monitors, ventilators, infusion pumps, etc.) in charge of the patient's vital support in Intensive Cares Units (ICU). The details of the components of the ICU room PC architecture, being part of the global SAIP system, have been shown. It has been necessary to develop communications drivers with a common interface to facilitate the connection to the different devices in a uniform way, and the acquisition of the data provided by the medical devices following the indications of the medical staff responsible for the ICU. These communication drivers have been developed as plugin libraries to allow the SAIP application to manage them through an agent entity, while only those corresponding to medical devices connected to the patient are loaded and used.

On the other hand, once the data of the medical constants of each device are gathered and transformed into VITAL nomenclature, they are stored in the MDIB of the corresponding ICU room PC. This allows to visualize as graphs (trend curves) the values stored for the different medical devices. These graphs allow the medical staff to see the evolution of the different medical signs provided by the devices during a complete cycle of 24 hours, or the ones stored for any day of the stay of the patient in the ICU.

Considering the implementation and the behaviour of this part of the system, the benefits and remarkable advantages related to the handling of information provided by the medical devices are:

- Automatic and periodical collection of values of vital signs, avoiding the manual introduction of these values in the system.

- Automatic collection of the alarms produced by these devices and diffusion for the knowledge of the medical staff of the ICU service.
- Storage of all this information. Capacity to consult data of different stages from the beginning of the stay of a patient. Usually medical devices do not have the capacity to store the information generated during all the stay of the patient in the ICU. Some of them have buffers of storage, but they are not big enough to store all the information relative to the complete entrance of the patient.
- Plug-and-play functionality, that simplifies the use of the system for communicating with the medical devices, specially for the nursing staff.
- Automatic collection of the constants indicated in the treatment, always with the corresponding validation of the nursing staff. Obtaining of nursing reports relative to these constants.
- Taking of values for semi-automatic generation of balances.
- Generation of trend curves.
- Compilation of the information in a unique and standard format that facilitates its integration and handling and provides a real interoperability among devices.
- An application that facilitates a joint visualization for the generation of diagnoses with all the information of several medical devices.

As it can be seen, the main purpose of the system is to facilitate the collection, storage and subsequent processing of the information provided by the medical devices connected to patients in the intensive care units in an homogeneous and standard way. The automation of the collected data has been possible to make it available to the medical staff in a friendly way, and a real interoperability among heterogeneous medial devices has been obtained.

ACKNOWLEDGEMENTS

This work has been funded by the SACYL (Sanidad Asistencial de Castilla y Leon - Health Welfare of Castilla y Leon), through agreement of collaboration between the University Clinical Hospital (HCU) of Valladolid and the Centre for Automation, Robotics and Information and Manufacturing Technology (CARTIF Foundation). We appreciate the collaboration of the intensive medicine staff of the HCU and the CARTIF development team involved in the project.

REFERENCES

(2007). Open soruce home page. URL: http://www.opensource.org. Last visit: june 2007.

(2007). Trolltech home page. URL: http://trolltech.com/. Last visit: june 2007.

Adhikari, N. and Lapinsky, S. E. (2003). Medical informatics in the intensive care unit: overview of technology assessment. *Journal of Critical Care*, 18(1):41–47.

Case, J., Fedor, M., Schoffstall, M., and and, J. D. (1990). RFC 1157 - Simple Network Management Protocol (SNMP). Technical report, Network Working Group. URL: http://www.faqs.org/rfcs/rfc1157.html. Last visit: june 2007.

CEN (2000). ENV 13735:2000. Health informatics - Interoperability of patient connected medical devices. European prestandard. English version.

CEN (2002). CEN/ISO ENV13734. Health informatics. Vital signs information representation. European prestandard.

CEN/CENELEC (1993). IEC 60601-1. Medical electrical equipment. General requierements for safety. European standard.

Fraenkel, D. J., Cowie, M., and Daley, P. (2003). Quality benefits of an intensive care clinical information system. *Critical Care Medicine*, 31(1):120–125.

Frize, M., Trigg, H., Stevenson, M., and Solven, F. G. (1997). Decision-support systems for critical care. In *American Medical Informatics Association Conference*.

Jose, S. S. (2007). Modelado y desarrollo en software libre de un sistema de analisis integral de pacientes en unidades de cuidados intensivos. In *FSWC 2007: Free Software World Conference 3.0*.

Leinwand, A. and Fang, K. (1995). *Network Management, a practical perspective*. Addison Wesley, London, 2nd edition.

Martin, S., Gutierrez, S., Jose, S. S., and Barrientos, F. (2004). Sistema de ayuda al analisis integral de paciente en unidades de cuidados intensivos (SAIP). In *INFORMED 2004, X Congreso Nacional de Informatica Medica. El tratamiento de la informacion, instrumento de valor aadido en la practica medica*, pages 185–192. SEIS, Sociedad Espaola de Informatica de la Salud.

Scherrer, J. R. and Spahni, S. (1999). Healthcare information system architecture (HISA) and its middleware models. In *AMIA Symposium, 1999 - amia.org*. AMIA: American Medical Informatics Association.

Shortliffe, E. H. (2001). *Medical Informatics. Computer Applications in Health Care and Biomedicine*. Springer, New York, 2nd edition.

Stallings, W. (1999). *SNMP, SNMPv2, SNMPv3, and RMON1 and 2*. Addison Wesley, London, 3th edition.