Patient Data Management System Medical Knowledge-Base Evaluation

Joseph Kairouz
B.Eng (Concordia University, Montreal, Canada), 1993

Department of Electrical Engineering
McGill University, Montreal
July, 1996

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Master in Electrical Engineering

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-612-19870-7

Canada

ABSTRACT

The purpose of this thesis is to evaluate the medical data management expert system at the Pediatric Intensive Care Unit of the Montreal Children's Hospital. The objective of this study is to provide a systematic method to evaluate and, progressively improve the knowledge embedded in the medical expert system.

Following a literature survey on evaluation techniques and architecture of existing expert systems, an overview of the Patient Data Management System hardware and software components is presented. The design of the Expert Monitoring System is elaborated. Following its installation in the Intensive Care Unit, the performance of the Expert Monitoring System is evaluated, operating on real vital sign data and corrections were formulated. A progressive evaluation technique, new methodology for evaluating an expert system knowledge-base is proposed for subsequent corrections and evaluations of the Expert Monitoring System.

RÉSUMÉ

La présente étude vise à évaluer le système expert de gestion des données médicales à l'unité des soins intensifs de l'Hôpital de Montréal pour Enfants. L'objectif de la recherche est de développer une méthode systématique pour l'évaluation et l'amélioration progressive des connaissances contenues dans le système expert.

Une revue de la littérature des techniques d'évaluation et de l'architecture des systèmes experts existants est présentée suivie d'un aperçu sur les composantes du système. La conception du système expert de surveillance est par la suite élaboré. Suite à l'implantation du système dans l'unité des soins intensifs, sa performance a été évaluée avec des données réelles suggérant ainsi certaines modifications nécessaires du système de gestion. Une nouvelle méthodologie d'évaluation progressive est alors suggérée pour des évaluations et des corrections ultérieures du système de gestion.

Acknowledgments

I would like to thank Professor Alfred Malowany, thesis supervisor, for his assistance throughout this study, Mr. Franco Carnevale and Dr. Ron Gottesman of the Montreal Children's Hospital for their help, the CEFRIO organization for their funding and the entire PDMS team.

My special thanks to my family, and especially my sister Sylvia for her encouragement, help and support.

Table of Contents

List of Tables

List of Figures

1. Introduction

The workload of the health-care professionals at the Intensive Care Unit (ICU) of the Montreal Children's Hospital (MCH), and the highly stressful environment in which they operate can lead to errors in their written or oral reports. In order to reduce the health-care professionals' workload, it is necessary to implement a medical data management and expert system which will help medical professionals in their decision making.

Presently, every nurse has to keep data up to date in a written format. They are required to plot and estimate the patient vital signs every half an hour based on the display of the bedside monitor, and to write short comments about the patient condition. Moreover, at the beginning of every shift, the nurse in charge of a patient gives an update of the patient's condition to the incoming nurse. Here, a computer system can provide the health-care professionals with tools to store, retrieve, present and analyze complex patient data thus reducing the nurse administrative workload. Moreover, such a computer based system could provide further assistance to medical professionals in diagnosing a patient's condition critical situation.

The Patient Data Management System (PDMS) is a computerized medical system that provides the ICU staff with automated data collection, storage and display, and generates a diagnosis of a patient's condition using an expert system. An important step in the PDMS product life cycle is to evaluate the system with regard to the accuracy of the diagnosis that it is able to generate. Thus, the expert system evaluation plays a key role in the implementation process of the intelligent medical system. Any inaccuracy may contribute to a bad patient outcome that could mislead doctors in crucial decision making situations.

The purpose of the present study is to evaluate the PDMS medical expert system. An emphasis is put on the elaboration of various methods to evaluate and correct the medical knowledge on which the expert system is based.

This chapter introduces the expert systems technology to the reader by giving a brief definition and overview of the a general case expert system. A general architecture and the programming paradigm involved in expert systems is presented, followed by a survey of different expert system evaluation method currently used.

1.1 Expert Systems

Expert systems are the first attempt of artificial intelligence to mimic the human behavior by modeling the world into a sequence of conditional statements. An expert system is "a computer program using expert knowledge to attain high level of performance in a narrow problem area" [Waterman, 1986]. Expert systems were developed in the Artificial Intelligence (AI) laboratories in order to provide effective services when applied to real life [Buchanan, 1984]. They are currently applied in various areas such as space, finance, communication, military systems, and medical systems, covering a wide variety of topics. Some of them involve processing natural language by using the syntactic structures of the English language, or knowledge representation and organization, trying to give the computer a structured representation similar to an organization by the taxonomy present in the human memory [Potter, 1990]. In general, expert systems are involved in problem-solving or decision-making. Stefik [Stefik, 1982] and later Waterman [Waterman, 1986] categorize expert system applications into ten groups exhibiting synthetic role where the expert system generate new information from the provided data, analytic role where the expert system provide interpretation and analysis of the data, or both synthetic and analysis roles:

1. interpretation: data analysis.
2. diagnosis: medical evaluation of a condition.
3. monitoring: continually interpreting signals and generating alarms, when required.
4. prediction: forecasting the course of the future based on a model of the past and the present.
5. planning: generating a plan of action to achieve goals required.
6. design: creating specifications to make objects satisfying some requirements.
7. debugging: identifying malfunctions in a process.
8. repairs: correcting identified malfunctions.
9. instruction: diagnosing, debugging and repairing novice behavior.
10. control: managing system behavior.

In the following, an overview of expert systems and their evaluation are provided.

1.1.1 Definition

When a goal is set, the problem of finding the way to attain it arises. It is conventional, for us, to think of goals and event sequences as metaphorical paths leading from one state to another. We speak of *searching for a solution, getting around road-blocks, getting lost* in the middle of a solution, *hitting a dead-end* and being forced to *back-track*. We also mention, in our expressions, *approaching the problem from a different angle*.

In the human brain, such processes of problem-solving and decision-making are conceived as searches in a metaphorical space, that underlines every knowledge understanding [Holyoak, 1989]. Similarly, expert systems are goal oriented software able to solve problems in a human-like fashion. Consequently, as intelligent systems, they differ from conventional computer programs in four ways:

1. They solve complex problems. For a person to solve the same problem, training and expertise in the domain are required.
2. They can formulate and then code the problem as an algorithm, which can not be done using conventional software.
3. They separate the expertise (knowledge base) from the mechanism applying the expertise (inference engine).
4. They use the reasoning process (or searching for a solution in a variety of possibilities) which is based on methods believed to be used by human experts (heuristic searches).

The knowledge can be encoded in different ways: rules, frames, semantic nets, and others. Nevertheless, the rule-based systems, also called production systems, dominate the industrial market[Rich, 1991]. These systems are based on predicate calculus. The rules are typically in the form of *if P then Q*, if P is true then Q is inferred. They are more easily accepted by users despite their discrepancy with the human reasoning [Fox, 1990]. In some situations, when Object Oriented programming is applied to expert systems, frames and semantic nets seemed to work best with this technology.

For expert systems, an evaluation is performed in order to determine (1) if the system accomplishes satisfactorily the task it is created for, and (2) if it contains the correct and representative knowledge of the domain. Nevertheless, the evaluation process of Expert Systems differs from that of software engineering and other domains. While both evaluation tasks aim to ensure a good product, there respective goal is essentially different. The first evaluation is concerned with the **verification** that checks the internal correctness of the product. It is defined as "building the system right", meaning, building the system correctly. The second evaluation known as the validation, checks the output correctness and accuracy that the expert system produces. It is defined as "building the right system", meaning, building the system that conforms to the specification of the product [Boehm, 1976].

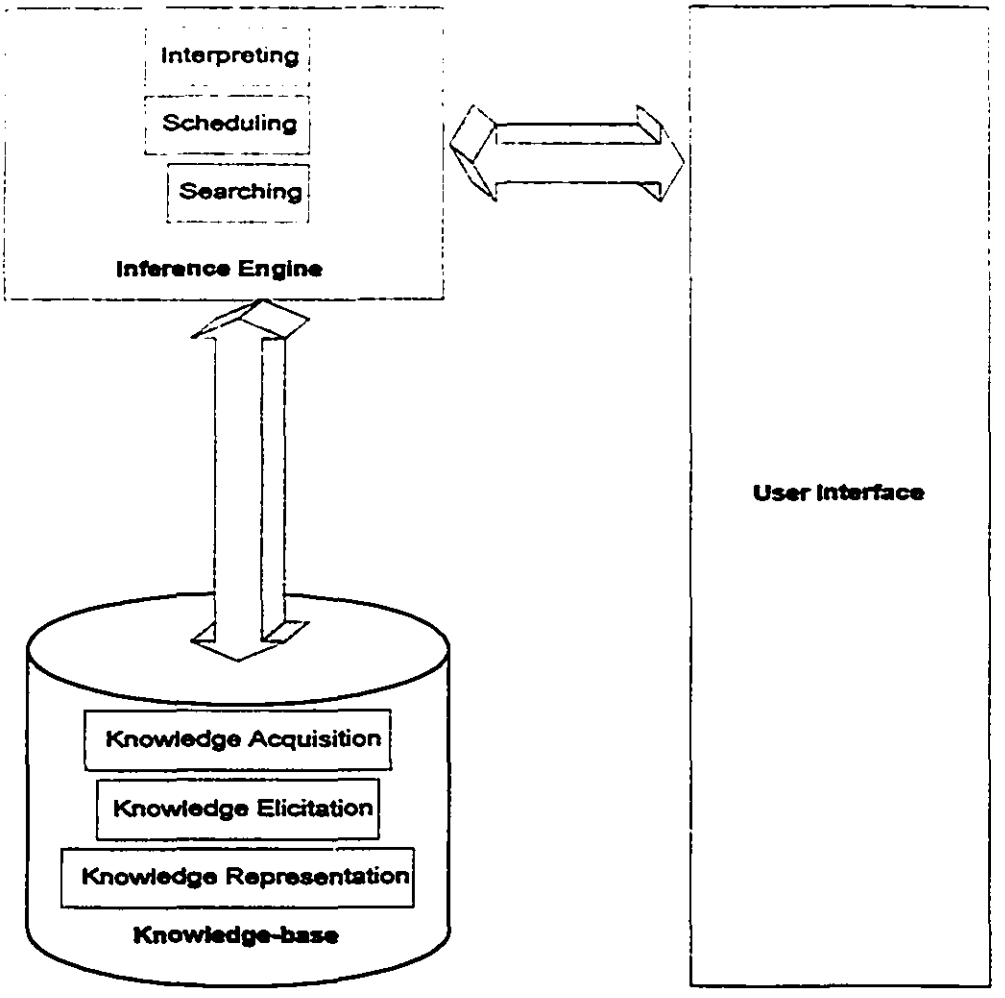

Figure 1 Expert system architecture

1.1.2 Expert systems architecture and programming

It is important to state that the expert systems software differ from traditional software in terms of architecture and life cycle. This section presents the architecture and life cycle of a typical expert system.

1.1.2.1 Expert system architecture

There is no universal implementation for expert systems although, one can identify a common architecture in expert system design (see Figure 1). Expert systems consist of a typical structure that includes: (1) the knowledge-base, (2) the inference engine, and, (3) the user interface. This three component structure enables the expert system to achieve the required tasks in a human-like fashion.

1.1.2.1.1 Knowledge-base

The knowledge-base is a database that contains pertinent information, facts that are collected from field experts, text books and other sources, expressed as conditions, objects representing reality or contingency topics. In expert systems, the database is composed of symbolic elements instead of literate or numeric elements like in other systems. This difference characterizes the structural configuration of expert systems.

The business of the knowledge collection is called knowledge engineering. It is achieved using specific defined steps, namely:

- Knowledge acquisition: it is the process of acquiring *knowledge* from field experts and other sources.
- Knowledge elicitation: it consist of coaxing *information* from human experts.

Rules are the most commonly used representation in knowledge encoding, although other types of knowledge representation exist (i.e. frames, cases).

1.1.2.1.2 Inference engine

The Inference engine is the main component of the expert system. It acts like a control structure and reasoning mechanism for the system. It is responsible for the general problem solving exercise, which consist of:

- Interpreting; i.e., analyzing and processing the rules.

- Scheduling; i.e., determining what to look at next.
- Searching; i.e., searching a limited portion of the knowledge to solve the problem by using heuristic searches.

The inference engine uses the knowledge-base to find solutions based on the user or the system input.

1.1.2.1.3 User interface

The user-interface is the link between the expert system and the end user. It is responsible for collecting information from its end-user and displaying results produced by the expert system. One expects the user interface to be graphical, user friendly and exhibit some kind of intelligence (i.e. color coding, speech recognition).

1.1.2.2 Expert system life-cycle

Just as expert systems structure differs from traditional software structure, so does the life-cycle. The expert system does not utilize the traditional *Waterfall* model [Lucas, 1986]. It is identified by three major activities [Liou, 1990]:

- Identification: It is the knowledge-base development.
- Design: it consists of designing the necessary mechanism that will produce the problem solving ability of the expert system.
- Formalism: it consists of transforming the collected knowledge during identification into machine readable knowledge, capable of being used by the system.

Finally, expert system shells are off-the-shelf products that provide the formalism in an expert system design.

1.2 Expert System Evaluation

One of the weaknesses of expert systems is the fact that there is no evident way for evaluating them. Expert systems introduced a new paradigm that differed from regular programming, not only in terms of architecture, but also in the type of resources used. The simultaneous introduction of a knowledge-base and reasoning is mainly what makes expert systems different from conventional programming. Thus, this innovation made the evaluation of the product more difficult and complex. As both regular programming and knowledge engineering were combined to provide the user what appears to be an intelligent behavior, there is no standard approach to evaluating an ES. Behind the intelligent behavior, more complex issues are hidden to the eye of the user. The expert system designers have to account for several parameters like performance, user friendliness, knowledge correctness and others factors that will give the expert system its potential to solve complex problems.

Three approaches were adopted for evaluating this mixed environment: (1) qualitative, (2) quantitative and (3) hybrid approaches. All these approaches consider the life-cycle of the product as a key role in the evaluation. Still, none of these approaches presents an evaluation that is complete enough to ensure the expected results (both performance and reasoning) from expert systems. In the following, a description of the qualitative, quantitative and hybrid approaches is presented.

1.2.1 Qualitative evaluation

Qualitative evaluations found their route in the work of Boehm [Boehm, 1976] who was a pioneer in covering the topic of software quality in 1976. The first qualitative evaluation that added value to the artificial intelligence domain was introduced in 1950 by Turing [Turing, 1950]. The Turing test got around evaluating the intelligence of a machine

by using people's common sense. The test consisted of making an operator formulate questions to be answered by an intelligent machine and by a person without knowing which is supplying the answer. If the operator is unable to distinguish between the machine and the person, then the machine is judged intelligent. Despite the fact that this test introduced new significance to artificial intelligence evaluation, it has pitfalls that made its virtues limited. The presence of an operator as absolute judge of the abilities of the machine diminishes the credibility of such a test. If the operator has some knowledge about intelligent machines, she or he might be able to tell if a machine or a person is on the other end of the evaluation.

Many researches adopted the Turing test or a modified version of the Turing test as a qualitative evaluation process [Gashnig, 1983; Hollnagel, 1989; O'Keefe, 1987; O'Leary, 1990; Turban, 1988]. They highlighted another limitations of the test: the test evaluates the end result of the machine (meaning the decision chosen by the intelligent machine) and not the different aspects of the process by which it generates the answers. Thus, it is impossible to standardize such an evaluation scheme.

Despite all the criticisms made toward the Turing test, it is still the most used approach for evaluating intelligent machines. Usually, researches use modified versions of this test.

Sharda *et al.* proposed a different method of evaluating intelligent machines, namely decision support systems. The idea is to take two different groups working on identical cases, one working with a decision support system and the other without any support system. The evaluation of the support system consist of evaluating different criteria of the end result, such as time consumed on the task or confidence in the result. The pitfall of such a test resides in the variations that can be introduced by the choice of the groups and the choice of the cases that are evaluated by the groups [Sharda, 1988].

Some authors introduced various checklists and guidelines to evaluate the qualitative side of expert systems. They dictate different methods to design, implement, develop, analyze and support expert systems. Gashnig [Gashnig, 1983] presented a list of criteria for expert system production. He proposed an evaluation based on efficiency, cost

effectiveness, hardware environment, discourse, decision, advice and performance. Others followed the example by adding elements to the previous list.

In all the evaluation schemes that have been considered above, the essence of expert systems, the knowledge-base, is considered to be one element among the others and does not play the major part in the evaluation. It is often not explicitly considered. The developers of the evaluation processes relied on the end-product to reflect the efficiency of the knowledge-base. Also, this kind of evaluation process does not take into consideration the scope and limits introduced on the system by the designer and the architect during the product development.

1.2.2 Quantitative approach

The object behind quantitative evaluation is to express the value of a system in terms of numeric measures of merit known as metrics. Formulas to evaluate expert systems are borrowed from different domains of software evaluation. In the following paragraph, we present several techniques of quantitative evaluation of expert systems.

McKerrow [McKerrow, 1988] presented the following criteria to measure quality in software physics:

- Reliability = $\frac{MTBF}{(1+MTBF)}$ where MTBF represents Mean Time Between Failure.
- Availability = $\frac{MTBF}{(MTBF+MTTR)}$ where MTTR represents the Mean Time To Repair.
- Maintainability = $\frac{1}{(1+MTTR)}$

Other techniques use metrics based on the number of errors present in a program. By deliberately placing errors (bugs) in the software, the system was evaluated according to the following formulae:

(number of errors uncovered/number of errors in the system) = (number of seeded errors uncovered/number of seeded error placed) .

Many researchers used probabilities, software modeling and statistical analysis to determine the quality of the expert system or decision ability and software quality. Hollnagel *et al.* modeled the system based on the probability of execution of a function, the probability of data triggering errors and if those errors were noticeable; results were produced utilizing differential equations [Hollnagel, 1989]. O'Keefe *et al.*, O'Leary *et al.* and Sackson *et al.* used statistical tests such as variance, correlation coefficients, confidence intervals, and stability measurements to compare expert systems and human experts abilities to generate solutions [O'Keefe, 1987; O'Leary, 1990; Sackson, 1990].

1.2.3 Hybrid approach

Many researchers tried to combine both the qualitative and the quantitative approaches to overcome the limitations of each technique. The advantage of a hybrid evaluation scheme is to blend the **common sense** of the qualitative evaluation to the **objective judgment** of the quantitative evaluation. Bailey and Pearson introduced the user-Information Satisfaction (UIS) as a subjective assessment in system evaluation [Bailey, 1983]. This hybrid approach was followed by Ives *et al.* [Ives, 1983], Baroudi and Orlinkowski [Baroudi, 1988], Doll and Torkzadeh [Doll, 1988], Galletta and Lederer [Galletta, 1989], and Rai and Mendellow [Rai, 1989].

O'Keefe [O'Keefe, 1989] proposed a multi-criteria method for assessing the decision making of decision support systems. Liebowitz [Liebowitz, 1986] proposed a hybrid evaluation based on an analytical hierarchy process. The evaluator prioritized the evaluation criteria reflecting a measure of goodness.

1.3 Thesis Overview

The objective of this thesis is to introduce a progressive evaluation methodology to the knowledge-base of the Patient Data Management System expert system of the Montreal Children's hospital. In chapter two, an overview of the PDMS software and hardware is given followed by the description of the PDMS medical expert system knowledge-base evaluation. Chapter three introduces and applies the evaluation model to the PDMS knowledge-base. Finally, chapter four describes the implementation of the evaluation process to the PDMS expert system software.

2. PDMS System

This chapter presents the Patient Data Management System (PDMS) of the Pediatric Intensive Care Unit (PICU) of the Montreal Children's hospital (MCH). First, the pertinence of building such a system is stated. The PDMS hardware is presented including a presentation of all the hardware required in the ICU to perform the vital signs data collection and the network required to communicate the data to the PDMS hardware. The different modules that constitute the PDMS software are presented. An in-depth description of the Expert Monitoring System module's conceptualization and design is presented; it is the module that contains the expert system responsible for diagnosis the patient's condition.

2.1 PDMS Objective

The development of medical information systems started in the early 1980's. The amount of data generated for every patient made it very hard for the nurses to faithfully record vital signs data. The need for computerized systems to perform the nurses administrative work faster and more accurately became apparent.

2.1.1 Situation in the ICU

Every patient of the ICU is connected to a bedside physiological monitor measuring all or some selected vital signs. Each bedside physiological monitor is linked to

various transducers placed on the patient. Even though they are able to display a graph for each measured vital sign, the bedside monitors do not have the capacity to store the generated data. It is the nurses responsibility to record the parameters every thirty minutes, to plot a graph based on an approximate mean value of the covered period of time and to keep track of the patient's condition. During each shift, data is written conjointly with a brief patient condition update by the nurse in charge of a patient. This is transmitted to the nurse responsible for the patient in the following shift.

2.1.2 Limitation of the system

The current non-computerized system that is used in the ICU is satisfactory but not optimal. In the actual work environment, nurses are frequently faced with executing two conflicting tasks simultaneously: the administration of the medical information and the patient care. Each nurse has to take care of the patient and to monitor the health condition, by keeping records of the vital signs, by updating graphs and by relaying information properly when the working shift ends. Both tasks need the nurse's attention which makes it hazardous during emergency cases. Moreover, this multifunctional-tasking that is required from nurses can introduce bias in the encoding and relaying of the data and ultimately alter the quality of the care that is provided.

Further, the representative parameter plots that are manually generated may not give the doctor sufficient information for her/him to make a thorough decision, independently of the nurse. The PDMS is proposed as a partial and possible solution to relieving the health care professional workload.

2.1.3 The PDMS, a solution to the problem

The PDMS is an ongoing joint research program involving McGill University's department of Electrical Engineering and the Montreal Children's Hospital. It is a computer-based real time medical information system that provides health care professionals with tools that enable the acquisition, management and manipulation complex patient data. The data is collected either, automatically, from physiological monitors, or input manually by a nurse in the case of parameters, such as fluid balance measurements or laboratory test results (the laboratory test results cannot be read automatically because of a lack of integration between various internal networks in the hospital). This system remedies some of the mentioned limitations inherent to the manual data processing systems by:

- Minimizing the risk of errors in encoding the patient condition
- Enabling the nurses to give a debriefing of the patient situation without relaying a large amount of information which it is already stored in the system and readily available
- Giving more time to the nurses to carry out their primary responsibility, which is, to take care of the patient

We consider that automating the administrative process would improve the current system in the following ways:

- It accelerate the data acquisition, manipulation and archiving.
- It offers a user-friendly interface with fast data review and interpretation in a window-based environment.
- It enhances the storage quality, the precision of the measurements, and the sampling rate from one reading every 30 minutes to one reading every minute.

In addition, a medical expert system provides further assistance by acting as a critical situation warning instrument and by providing support to medical domain professionals in their decision making. The user's tasks are substantially simplified; they

consist of complementing check lists or legal form documents using the keyboard or the mouse. All needed information is presented using narrative style, or color coded icons

Thus, the PDMS reduces the nurses' administrative work, allows a better patient monitoring, and, it helps doctors in their decision making by (1) warning them of any critical situations, (2) offering a diagnosis of the patient medical condition and, (3) allowing an accurate review of the dynamics of the patient vital signs [Kairouz et al., 1994].

2.2 PDMS Hardware Architecture

The PDMS hardware requirements are simply an IBM compatible Personal Computer. A HP medical network is also required to collect the vital sign data from the patient and make them available over the network to the PDMS. This chapter presents the hardware configuration used in the hospital ICU and the hardware environment used in the McGill laboratories to develop the PDMS software.

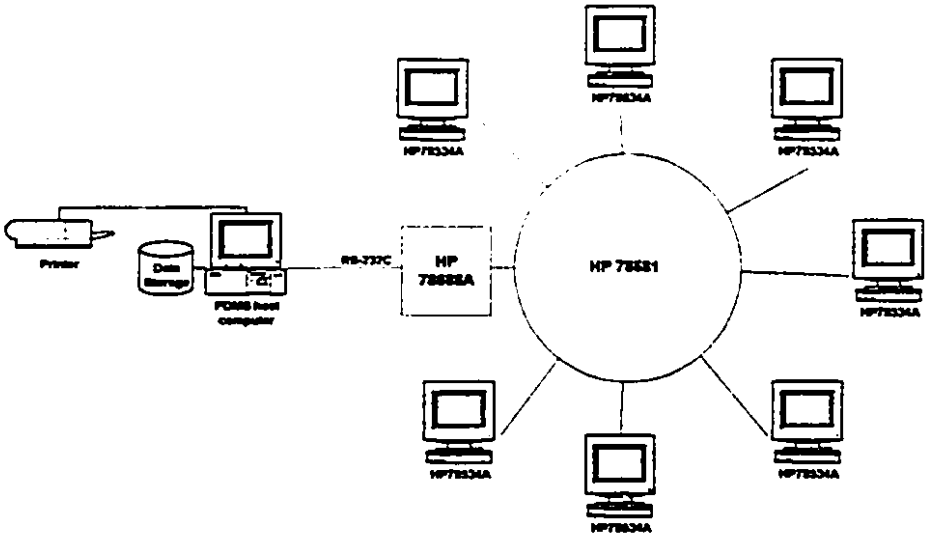

Figure 2 The network setup in the ICU

2.2.1 Material

The main component of the PDMS is the Hewlett-Packard HP Care-Net medical system. Fourteen HP78534A Bedside Physiological Monitors/Terminals are linked to HP78581 Network System Communication Controller. The PDMS host computer system is linked to the Network System Communication Controller via the HP78588A Careport Network Interface by an RS-232C serial line, see Figure 2.

2.2.2 Characteristics

The HP Care-Net is a real-time medical information system that offers the capability of networking different medical devices into a proprietary Local Area Network (LAN). The HP LAN offers the following options:

- **Duplex communication**: a two way communication that allow the user to send and receive information from the patient monitor screens.
- **Network robustness**: A star topology is used to link the fourteen Bedside physiological Monitors/Terminals and the different networkable medical devices to the Network System Communication Controller which acts as a central controller. In this topology, all nodes are connected to the central node via point-to-point links. Thus, a branch failure does not affect the performance of the network, and the branch failure is sensed and isolated from the network by the central controller. The disadvantage of such a geometry is that in case of a central node failure the whole network will be down.
- **Real-time processing**: the central controller uses a polling communication protocol with fixed maximum duration and token passing protocol within the polling cycle. This insures a real-time data transmission over the network.

2.2.3 Communication mode

The HP78581 Network System Communication Controller is the active node of the HP Serial Distribution Network (SDN). It provides the physical communication link to all the nodes connected to the LAN. It accommodates up to thirty two nodes, twenty four may connect to HP bedside instruments, six may connect to patient information centers, and two may connect to computerized monitoring and management systems.

The HP78534A Bedside Physiological Monitors/Terminals measure the different parameters selected by the user and send the result to the central node.

The HP78588A Careport Programmable SDN Interface is the link between the PDMS host computer and the SDN. The Careport can acquire four types of SDN data: parameter data, waveforms, bedside alarms and inops (inops are signals generated by disconnected transducers), and instruments status. The data acquisition read rate is programmable.

The PDMS host computer, containing all the PDMS software modules, is a Personal Computer (PC), Intel 80486, with 16 Megabyte Random Access Memory (RAM) and 200 Megabyte hard disk total space. It is running under IBM OS/2 operating system, version 2.0. It uses a high resolution color display adapter providing a 1024X768 pixels resolution. An RS-232C serial line connects the host computer to the HP Careport.

2.2.4 Lab configuration

At the McGill laboratories, a different configuration is used to develop and test current and future implementations. Two IBM PS/2 model 80 Intel 80386 PCs with 8 Megabytes RAM are used, one of the PCs simulates the medical network and the other PC act as the PDMS host computer. Simulation data generated in one PC is sent over a serial link to the second PC running the PDMS software.

2.3 PDMS Software Architecture

After a description of the hardware, we present in this section the architecture of the software that supports the PDMS. The PDMS software is constantly being improved and updated in McGill University's Electrical Engineering laboratories. In this section, the

choice of the operating system will be discussed followed by a brief description of each of the seven software modules. This presentation will facilitate the understanding of the evaluation process of the system.

2.3.1 The PDMS operating system

Originally, the PDMS was developed under the MS-DOS operating system. It consisted of collecting patient vital signs and displaying them by using a character-oriented display on a monochrome screen. In 1989, a new PDMS design was proposed offering a revised and improved user-interface based on the Window-Icon-Mouse-Pointer standard. At that time, the MS-DOS did not offer support for a graphical interface. IBM OS/2 operating system offered the capability to provide the user and the designer with a windowing graphical interface that allows (1) the development of a friendly user-interface, (2) the use of modular system design, and (3) the use of extended memory management in order to compensate for the increasing complexity of the PDMS. Currently, new PDMS modules are being developed under IBM OS/2 Warp.

OS/2 is an affordable operating system that can be implemented on a PC platform. It supports multitasking, resource management, large real memory, virtual memory, memory isolation, I/O protection and can execute software written under MS-DOS OS. The PDMS was implemented in this environment. The following sections describe the various software components of the PDMS and their respective functions in the system.

2.3.2 The PDMS modules

The PDMS software is broken into a collection of modules. In an ongoing research where modifications and add-ons are common practice, modularity seems to be the best solution for software evolution. It offers flexibility in the design and simplifies the coding

complexity by breaking down the program into a collection of distinct tasks that can be implemented independently. Modules can be added or eliminated without affecting the others. Modularity also offers a protection: a corrupted module will affect the rest to a limited extend (or in a limited way).

The PDMS software was developed using C language, under IBM OS/2 operating system. The implementation exploits modularity by using the multitasking services and interprocess communication functions such as queues, shared memory, semaphores and named pipes.

The PDMS software is a collection of seven modules that are developed or currently under development. In the following sections, each of the PDMS modules and their respective functionality are described. These modules are the Data Link Controller, the Register Module, the Database, the Vital Signs Monitoring Module, the Fluid Balance, the Nurse Workload Manager, and the Expert Monitoring System. The evaluation developed in this study is for the expert system knowledge-base of the Expert Monitoring System module. Therefore, the Expert Monitoring System is presented in more detail than the other modules.

2.3.2.1 Data Link Controller

The Data Link Controller (DLC) module is responsible for the data communication between the host computer and the HP Careport. It automatically acquires the available data on the network in real-time and makes it accessible to all the PDMS system. The DLC obtains new data every two seconds; from that data, minute data and half an hour data points are created by averaging the second data. The data is then temporary stored in circular queues in a shared memory so it can be available to all the modules. Semaphore handshaking is implemented for parameter queues and network data access in order to avoid read/write conflicts.

2.3.2.2 The Registration Module

The Registration module is responsible for acquiring all patients administrative data and status. Information, such as name, sex, age, address, telephone number, bed number and others, are entered by the user through a menu-driven user-interface. Thus, the user can admit, suspend and discharge a patient at any time. In case of an emergency at the admission time, minimal information is required and the user can register the patient with as little information as bed number. Later, the patient information can be edited and modified.

While entering the different data requested by the registration module, error checking routines inform the user of a variety of errors due to mistype or inattention.

2.3.2.3 The Database Module

The database of the PDMS is currently under development. In its present design, the database creates tables for Patients Registration, Nurse Care Plans, Vital Signs, and the EMS patients conditions.

The PDMS database utilizes the relational database included in the OS/2 Extended Edition Database Manager. The module is written in C language with embedded Structured Query Language (SQL) statement type.

For the purposes of this evaluation, the expert monitoring results and patient vital sign data were stored in a flat file system.

2.3.2.4 The Vital Signs Monitoring System Module

The Vital Sign Monitoring System (VSMS) module is a graphical user interface to plot the patient data. It is a powerful visualization tool. It acts as a tool to be used by doctors permitting then to adjust the way they would like it to look at the plots. The effectiveness of the VSMS resides in the use of visual coding to help the user understand the totality of the gathered information effortlessly. Line color, marker color, line style and marker type are used to create this advanced graphical-user interface visual coding.

The VSMS allows its user to view as many vital signs as available for multiple beds simultaneously. If large sets of data are viewed, the VSMS provides the user with horizontal and vertical scroll bars [Yien, 1990].

2.3.2.5 The Fluid Balance with Speech Interface Module

The Fluid Balance (FB) module is responsible for monitoring the intake (ingesta) and the output (excreta) of the patient in a spreadsheet form. This module does not collect its data form the shared memory, but from a periodic readings of the infusion pumps or urine bags performed by the nurse. The data is then entered into an electronically reproduced fluid balance chart using a speech interface system.

The speech interface developed consists of a speech recognition system and a speech synthesis system. The speech recognition system translates into machine commands data entered into the computer through voice commands using a headset that provides a hand-free and eyes-free operation in order to enhance the nurse mobility while entering the data. The speech synthesis system eliminates possible errors by echoing back the inputted data for confirmation before committing it into the FB tables [Petroni, 1991].

2.3.2.6 Nursing Workload Manager Module

The Nursing Workload Manager (NWM) module is designed to automate the workload management of the nurses. It manages patient nursing care plans created manually or by calling up a standard care plan from a library and customizing it for the patient. An automatic scoring system was incorporated according to the Progressive Research in Nursing (PRN) workload measurement system. The NWM sets up the Fluid Balance charts through integration with the fluid balance module [Rogers, 1992].

The NWM module is also responsible for scheduling the nurses activities using an expert system. The input of the scheduler is limited to eight categories in the nurse care plan: respiration, elimination, personnel care, communication, treatments, diagnostic, nutrition and hydration procedure.

2.3.2.7 The Expert Monitoring System Module

The Expert Monitoring System (EMS) module is described in the following section. The evaluation of the EMS is the main topic of this document, therefore a more detailed description of the expert system is provided. The EMS can be defined as a medical tool and a decision support machine [Lam, 1992].

2.4 Expert Monitoring System

In this section, the concepts underlying the different parts that constitute the EMS module and the implementation of these parts will be described. An evaluation technique of the medical expert system decision making ability is then introduced and discussed.

2.4.1 Functionality

Before proceeding with a technical discussion of the EMS implementation, a brief overview of the medical context of the expert system will be discussed. The cardiovascular system, principally composed of the heart, the arterial, and the venous systems, generally reflects the medical condition of a human being. Unfortunately, this system is not governed by a simple mathematical equation. One can only derive an approximate mathematical description of the cardiovascular system using fluid mechanics theory. To date, it seems impossible to account for all existing parameters and their dynamic interactions. The modeling of the cardiovascular system lacks the exactness that allows a reliable description of a person's condition.

The Heart Rate (HR), Blood Pressure (BP) and Central Venus Pressure (CVP) closely describe the cardiovascular system status; they are the most monitored parameters in the ICU. The expert system of the EMS monitors, analyzes and interprets the aforementioned vital sign parameters in real time. It detects and predicts life threatening events, proposes a diagnosis of the patient conditions and generates warning signals. The expert system acts like a medical tool and a decision support machine.

2.4.2 Design and specification

Like the other PDMS modules, the EMS was written in the C language, using Microsoft C compiler, version 6.0, operating under IBM OS/2 version 1.3 or higher. The EMS software design consists of three different components: the linearization algorithm, the expert system and the graphical user interface. The EMS module runs in a real time mode. The data, acquired by the DLC module and the Registration module is stored in a shared memory. It is sampled by the EMS, at a rate of 1 sample per minute, from the minute circular queue (the minute data is an average of the second data). The data is then

processed in a sequential order passing through the linearization algorithm, the expert system, and presented to the user in a graphical format.

2.4.2.1 "Linearization[1]" algorithm

The linearization algorithm role is to minimize the measurement inaccuracy of the patient's parameters. The data gathered by the HP network is subject to a high level of "noise" due to patient activities such as coughing or crying. To compensate for such errors, the data is filtered. The raw minute data passes through a 2 points interpolation algorithm, producing a half minute data. The new half minute data is the result of the average of the current and the previous data. The interpolated data passes through a 17 then a 13 points median filter removing the unwanted impulses present and preserving the sharpness of the trend. Unfortunately, this comes with a cost since the two median filters can introduce, respectively, an 8-point and a 6-point delay, resulting in an overall worst condition of a 7 minutes delay.

Further details on the filter implementations can be found in [Lam, 93].

2.4.2.2 Expert system diagnosis of patient condition

The expert system was implemented using the Nexpert Object version 2.0 expert system shell produced by Neuron Data. In this section, the expert system mechanism used to generate a patient condition diagnosis is discussed. The knowledge-base design and the used certainty index technique are explained.

---

[1] The process referred to as linearization in the document does not describe a process leading to linear data in a mathematical way. It actually approximate data by a sequence of straight lines segments. The reader is to note that the some operation steps are non-linear by nature.

2.4.2.2.1 The EMS concept

The main part of the EMS resides in the knowledge-base and the rules of the expert system. As the EMS module is to execute in a real time mode, execution time is a major factor when creating the rules. Combined knowledge is used to write the rules, resulting in simple rules requiring small heuristic searches from the inference engine. Using this approach, the response time of the expert system is a function of the number of ICU patients being monitored by avoiding "deep" expert system searches for each case.

Table 1 Description of the eleven medical cases

Condition	Description
Agitation	Patient in a stressful situation due to pain or panic
Bradyarrhythmia	Abnormally slow pulse followed by a decrease of the arterial blood pressure and an increase of the venus blood pressure
CNP ICP	(Central Venus Pressure Intracrainial Pressure) All CNS abnormally regulate the nervous system of the cardiovascular system
CNP Drugs	Abnormal nervous system regulation due to drugs administrated to the patient
Pump Failure	The heart is not able to pump the blood correctly
Hypovolemia	A too low blood volume, due to a dehydration or severe burns.
Hypervolemia	Excess of blood volume
Primary Hypertension	A hypertension case causing changes in the cardiovascular system
Tachyarrythmia	Abnormally fast heart rate causing heart problems
Tamponade	Internal bleeding in the region of the heart, weakening the heart and causing it problems.
Systemic Shock	A chemical reaction causing an abrupt decrease of the blood pressure

2.4.2.2.2 Knowledge-base and rules

Based on the HR, BP and CVP, the most common patient conditions found in a pediatric ICU were selected. A brief description of these conditions is given in Table 1.

Table 2 Expert system rules

Heart Rate	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	+	+	+	+	+	+	+	+
Blood Pressure	-	-	-	-	-	-	+	+	+	-	-	-	-	-	-	+	+	+	-	-	-	-	-	-	+	+	+
Central Venus Pressure	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+	-	-	+
Hypovolemia	x			x		x	x			x			x			x			x			x			x	x	
Hypervolemia									x						x									x			x
Bradyarrhythmia	x	x	x	x	x	x																					
Tachyarythmia																			x	x	x		x	x			
Tamponade			x																								
Pump Failure	x		x							x		x									x						
CNS-ICP							x	x	x																x	x	x
CNS-Drugs										x	x	x				x	x	x									
Primary Hypertension							x	x	x								x	x								x	
Systemic Shock											x	x							x	x	x	x	x				
Agitation																						x	x	x	x	x	x
Normal													x														

Each of the conditions described in Table 1 is present in the expert system knowledge-base as summarized in Table 2. The patient vital signs (HR, BP, CVP) are classified in three levels: high, normal and low, represented in the table by +, = and - respectively. The "X" on the table mark the place where the condition is met for a given combination of the classified vital signs. Table 2 clearly demonstrates the combined knowledge present in the expert system knowledge-base.

2.4.2.2.3 Parameters classification

The vital sign parameters values are classified into different levels previously referred to as high, normal and low. The classification of the vital signs varies with the age of the patient. The condition of the patient starts deteriorating when one or more the vital signs diverge from its normal value. Manipulating the values of the parameters in a

classified manner could also be served using Fuzzy Logic. Unfortunately, the Nexpert expert system shell used, does not support the manipulation of fuzzy logic rules. Using the automatic rule manipulation in Nexpert, the above three level classification is further subdivided into five levels of classification of each parameter: critically high, alarmingly high, normal, alarmingly low, critically low, numerically represented by: +2, +1, 0, -1, -2. The five level classification introduces greater flexibility in the process and a finer discrimination.

Despite the fact that the expert system considers all three parameters to reach a diagnosis, only the worst case vital sign condition is used to determine the overall condition of the patient. The formula used by the expert system to account for the vital sign condition is described below:

$$d_x = \max(|HR_{condition}|, |BP_{condition}|, |CVP_{condition}|) \qquad \text{eq. 1}$$

where $2 < d_x < 0$

d_x Patient condition

HR heart rate

BP blood pressure

CVP central venus pressure

2.4.2.2.4 Certainty index technique

The technique used for evaluating the certainty index reflects the repeated occurrence of a diagnosis to give a continuous or smoothed evaluation in the diagnosis of the patient. This section describes how the certainty index achieves such a smoothing in the diagnosis.

Once the patient condition has been diagnosed according to the rules described earlier, the expert system can, simply, give this result to the user. But, when a doctor examines a patient, interpreting the instantaneous patient vital signs values is insufficient. The doctors also place importance on trending or the evolution of the patient's condition

in time. To address the trending aspect, a Certainty Index (CI) technique that follows the patient condition dynamics in time is implemented.

The CI accumulate based on a repeated occurrence of a diagnosis. The CI of a condition increases each time a patient condition diagnosis occurs. So if the vital signs of a patient change for a short period of time, the CI technique will prevent the estimated patient condition from jumping from one diagnosis to another. Instead, a gradual shift from is incorporated to stabilize or improve the medical judgment of the expert system in the presence of "noisy" vital signs.

$$\Delta CI_x = \begin{cases} CI_{max}, if CI_x > CI_{max} \\ CI_{min}, if CI_x < CI_{min} \\ w^r \sum_x (w_+ d^r_{x+} - w_- d^r_{x-}) + w^p \sum_x (w_+ d^p_{x+} - w_- d^p_{x-}) \end{cases}$$

-eq.2

where

$1 - w^p = w^r$

$1 - w_+ = w_-$

w^p weight of processed data

w^r weight of raw data

$x+$ positive diagnosis for condition x

w_+ weight of positive diagnosis

$x-$ negative diagnosis for condition x

w_- weight of negative diagnosis

d^r rule applied to raw data

d^p rule applied to processed data

x patient condition of Table 2, x is between 0 and 2

The current values of each patient condition (eleven in all) are updated from the previous estimation using eq. 2.

The CI vary between a minimum of -1 and a maximum of +1. A negative CI represents evidence rejecting the hypothesis of a diagnosis of condition *x* while a positive CI means that the vital sign evidence supports the diagnosis of this condition *x*. Determining the sign of the CI is based on Table 2. If the combination of the classified vital signs fire a rule, all the diagnoses that have an x for that rule have their CI increased, diagnosis with no x see their CI decreased. Both, data processed by the linearization algorithm (processed data) and the original data (raw data), are integrated calculating the value of the CI as shown in eq. 2. The non-filtered data is used to compensate for the delay introduced by the median filters. The CI of each condition is calculated using $CI_{updated} = CI_{previous} + \Delta CI$.

2.4.2.2.5 Certainty index prediction

In addition to offering the user a diagnosis of the present condition of the patient, the expert system uses the certainty index technique to offer a prediction of the certainty indices of the patient in a near future. For example, in the case of a blood pressure reading, categorized as alarming, and steadily rising, analysis of this trend predicts that the blood pressure will reach the critical category in a near future. This forms the basis of the strategy used to calculate predicted CI. Based on the latest trend analysis (current slope) of the HR, BP and CVP minute data, an extrapolation is made to calculate τ_p, the time required for each vital sign to cross into its next classified level as indicated in eq. 3.

$$\tau_j = \begin{cases} \dfrac{S_{p,j} - p_{position}}{P_{slop}}, if \dfrac{S_{p,j} - p_{position}}{P_{slop}} \geq 0 \\ \infty, otherwise \end{cases} \quad \text{- eq. 3}$$

$$t_i = \min|_j (\tau_j)$$

where $S_{p,j}$ next classification threshold level j in the direction of its slope

$p_{position}$ value of a parameter p at the current time

P_{slop} slope of a parameter p at the current time

τ_j calculated time for parameter p to reach its next classified level j

$$w_i^p(t_i) = \begin{cases} 1, if\, t_i \le 0 \\ 1 - \dfrac{t_i}{t_{max}}, if\, 0 < t_i \le t_{max} \\ 0, otherwise \end{cases} \quad \text{- eq. 4}$$

where w_i prediction weight for parameter p

t_{max} maximum allowed prediction time

$$d_x^p = \max(w(t_{HR})|HR_{next}^p|, w(t_{BP})|BP_{next}^p|, w(t_{cvp})|CVP_{next}^p|)$$

$$d_x^p = \max(w(t_{HR})|HR_{next}^r|, w(t_{BP})|BP_{next}^r|, w(t_{cvp})|CVP_{next}^r|)$$

where w weights for predicted calculations

$|parameter_{next}^r|$ next raw parameter classification 0, 1 or 2

$|parameter_{next}^p|$ next processed parameter classification 0, 1 or 2

The predicted CIx are then updated using w_i^p weights and d_x^p given in eq. 4. Again, the predicted CI are generated form the weighted combination using both raw and processed data.

2.4.2.2.6 Patient Condition

The certainty index prediction alerts the user of a potential complication or emergency. The overall condition of the patient is presented to the end-user in graphical format using green yellow or red color codes to represent the state of the diagnosed patient condition. The overall condition of a patient is derived from the eleven CI's

computed for that patient. The overall condition of the patient is assigned from the highest CI. A CI greater than 0.5 will lead to a yellow condition, red condition is given to a CI greater than 0.75.

2.4.2.2.7 Implementation

The EMS module, is composed of three main parts, filters, medical rules and the user interface. The PDMS shared memory is intensively used in the EMS module. Data is read from it and written to it. The EMS runs in real time, using its own asynchronous clock. It is set to cycle through a series of processes every minute. If it requires more time to complete the cycle, the next run time is postponed by a unit of one minute, for as many unit times as needed. This implementation gives flexibility to the EMS, allowing it to operate on slow as well as on fast computers. As a result, a faster computer updates the diagnosis every minute, whereas a slower computer does the updating every cycle. The cycle time, on an Intel 80386 based computer with fourteen beds registered can take up to three minutes.

The EMS reads the patients' vital signs, which are made available by the DLC in shared memory. From the data that is read, four different types of parameters are generated:

- The actual raw data which are the original vital signs of the patients collected from the HP network.
- The actual processed data corresponding to raw data processed by the linearization algorithm.
- The predicted raw data which are the predicted vital signs of the patient derived from the actual raw data.
- The predicted processed data which represents the predicted vital signs derived from the processed data.

The actual raw and the actual processed parameters are used to calculate the actual CI of the eleven proposed conditions and to give the current specific condition of the

patient. The predicted parameters are used to generate the predicted CI and condition of the patient.

Once the raw data is read, the EMS classifies each parameter as high, normal or low using look-up tables loaded in memory at initialization time. This three level classification is made according to the parameter type (HR, BP, CVP) taking into consideration the patient's age. Then, the classified parameters can be directly applied to the rules, using Table 2. To manage the application of the 27 rules on the four different types of parameter, the action of reasoning of the expert system is divided into six separate steps. These steps sequentially apply the rules on the parameters as described in the following:

Initialization: it informs the process to become active by a flag system, as soon as the data arrives.

Step 1 Read new data: the expert system checks the patient bed occupancy and reads the data of the registered patients.

Step 2 Run **diagnostics on actual processed data**: the 27 rules are applied to the actual processed data and CI are calculated for this data.

Step 3 Run diagnostics on actual raw data: the 27 rules are applied to the actual raw data and the CI are calculated.

Step 4 Run diagnostics on predicted processed data: the 27 rules are applied to the predicted processed data and CI are calculated.

Step 5 Run diagnostics on predicted raw data: the 27 rules are applied to the predicted raw data and the CI are calculated.

Step 6 Wait for new end data: all steps are completed, this is a wait state, the expert system is in idle mode, waiting for new data to come.

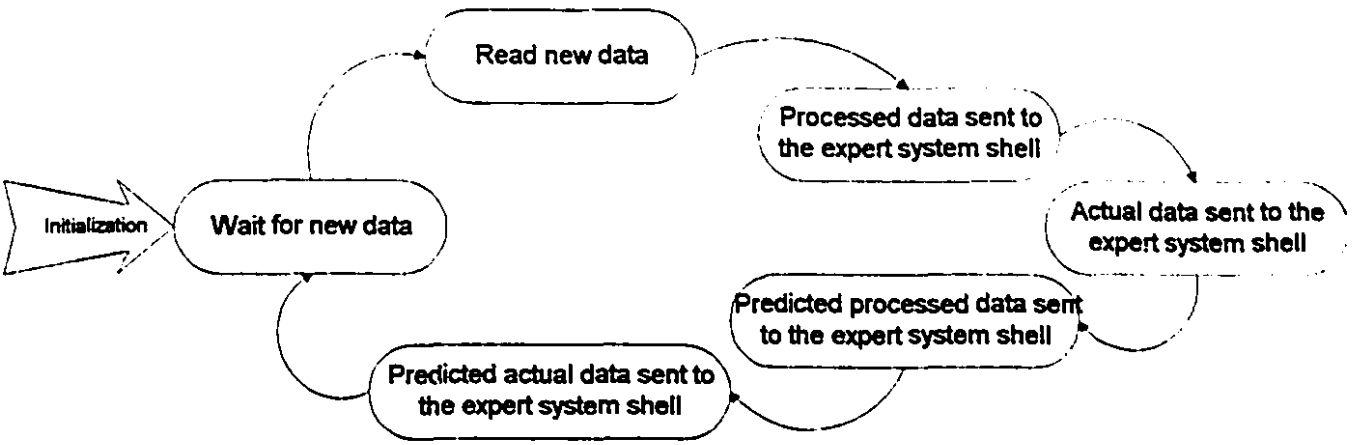

Figure 3 Reasoning state machine

Figure 3 shows the six steps that control the reasoning process. Actual data represent the instant data collected from the HP network, processed data represent the filtered actual data.

Once the cycle of the expert system is completed, the CI are written to the shared memory which is used to permit other modules or programs to access the new generated information. Currently, only the user interface of the EMS reads the CI from the shared memory to display it to the user. A diagram of the data flow involving the EMS in the shared memory is illustrated in Figure 4.

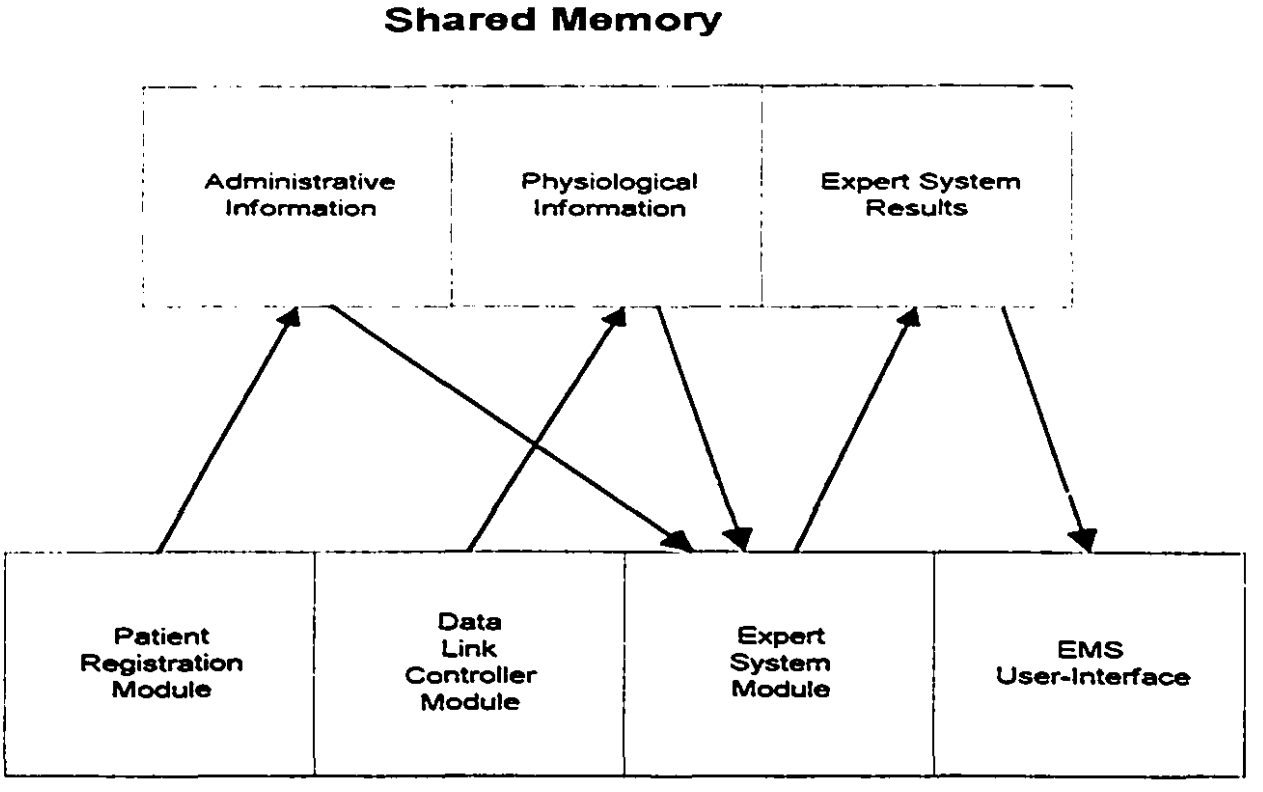

Figure 4 Internal system data communication

2.4.2.3 User interface

The user interface of the EMS reproduces the state of the fourteen beds of the Montreal Children's Hospital ICU on the screen. To present the actual and predicted condition of the patient, a coloring and bordering technique is used. The color of the bed represents the actual condition of the patient The predicted condition is represented by the color of a frame around the bed and the name of the patient. The normal, alarming and critical conditions are respectively represented by green, yellow and red. An empty bed is represented by a black box.

By clicking on a bed icon, the user can obtain more information about the registered patient. A window appears containing the name of the patient, the 22 CIs of the patient (11 actual CIs and 11 predicted CIs), the value of the three parameters used in the EMS (HR, BP and CVP), and a literal description of the patient condition. The user can go through all registered beds in the ICU by using this window.

IBM OS/2 Presentation Manager (PM) graphical user interface is used to create the user interface of the EMS module. PM window services are similar to the Microsoft Windows look under DOS with the advantage that most people are familiar with the Microsoft Windows environment. The PM Graphics Engine exploits the 32 bit flat memory model of OS/2 2.1, resulting in noticeably improved performance over the 16 bit OS2 version 1.3.

The user interface runs in an OS/2 session, separate from the linearization algorithm and the expert system since Nexpert Object version 2.0 is not designed with a graphical interface supporting OS/2 PM. Running a separate session for the user interface allows the user(s) to run multiple interface sessions concurrently.

3. Evaluation Process

The main objective that was initially pursued in the development of expert systems was to mimic human behavior in a specific problem solving domain. If one looks at human beings, this behavior can be described simply as a result of a system that is based on the human problem solving ability. In such a system, the input is the situation at hand, and the output is the decision or the solution that gives place to the behavior. A more comprehensive approach to the human decision making process reveals that other mechanisms are involved in the process along with the problem solving ability. We can distinguish processes like remembrance, categorization, judgment, choice, rationalization and others. All these human abilities are used in concert to produce the end-product of the process, that is the solution.

When one tries to formulate the knowledge of an expert into a form of knowledge representation, that is governed by some rules, the difficulty of such a task is far greater than the simple formalism of conditional situations. Such a system must be exhaustive in order to be efficient. Also, it lacks the capacity to learn and thus to generate new paradigms to serve new situations. Moreover, the system is not able to judge the validity of the decision. It evaluates exclusively the adequacy of the response relatively to the rules that have been prescribed.

It has been proven over the years that decision making can be formulated in a number of conditional statements. The difficulty is in correctly translating the knowledge that is accumulated from education, case studies, experience, reading, and common sense into these conditional statements. On top of this, once this knowledge is encrypted, it still does not serve the purpose of decision making the same way a person does. Once encrypted, the knowledge is not capable of learning and maturing from experience. The role of support seems to best describe this situation of stagnant knowledge. As a decision

support, the user can not expect the expert system to have an up to date knowledge all the time. Moreover, the user can not expect the system's decisions to be accurate all the time. Thus, the system can be improved only when an incorrect decision emerges. Then, it can be corrected and improved by a good evaluation.

The limits of intelligent machines that have been stated before suggest the importance to evaluate the knowledge on which the expert system bases the process of decision making.

The knowledge evaluation plays a key role in expert systems that is less essential in non-intelligent systems. By no means is it supposed to evaluate the intelligent aspect of the product, as there currently is no way to evaluate intelligence. The evaluation can act as the learning process of the system by pointing out mistakes and helping correct them. A progressive evaluation throughout the life of the product helps mature the system and provides a tested, corrected and enhanced reusable knowledge-base. The initial role of a progressive evaluation can be corrective. Once the first step completed, the evaluation role is to enhance the knowledge encrypted by minimizing the errors in of the decision making process.

Developing a process to evaluate an expert system involves a search for an effective questioning and testing of the system under verification and validation. The evaluation process has to be systematic, case independent, and it has to exhibit efficiency, meaning no redundancy in the information collection. This chapter (1) introduces the progressive evaluation technique, a systematic process for evaluating an expert system and (2) applies this process to the PDMS medical expert system, the EMS. The first part focuses on the process of information collection and on the development of an evaluation process. Although it describes the method at a high level, the second chapter gives a more thorough explanation of the method by providing an evaluation scheme specific to the PDMS.

3.1 Evaluation Objective

A complex product, such as an intelligent machine, whether it is a medical system, or an airplane repair system, encounters a variety of problems at evaluation time that leave confusion in the mind of the troubleshooter. The person correcting these flaws experiences uncertainty over where to begin, how to recognize problems that require action, how to break down confusing issues into simple components. Moreover, one has to set an order among the various issues according to their importance. The objective of this section is to introduce a systematic approach for complex product evaluation, independent of the evaluation method used (the evaluation method can be a questionnaire, metrics measurements or other means). This systematic approach considers the different issues as independent entities in a simultaneous and effective manner without creating confusion. This chosen approach creates an evaluation focused on problem solving and forecasting by showing:

- Where to begin: for complex systems, such as a medical expert system, the problems do not occur in an isolated fashion, they occur in large numbers. A starting point, dealing with these issues can sometimes be discouraging and disconcerting.
- How to recognize important cases; by nature, people tend to focus on secondary issues as they are simpler to understand and can be solved in shorter time. If not explicitly pointed at, important issues are left undone, as they are usually more difficult to understand.
- How to break the problematic cases into manageable components: most of the anomalies that can appear in complex systems are conglomerates of small problems. The bundle of problems have to be separated and dealt with independently. Separating the problem facilitates the search for a solution. The complexity of a problem is far greater, and sometimes makes it unresolvable, when it is made of a multitude of smaller problems.

- How to set priorities: recognizing the important problem of an issue is getting one step closer to the solution. Prioritizing allows the management of a problem and reduces the time needed to solve it.

This will help the evaluator to deal with the disorderly flow of information in situations that are invariably confusing, multifaceted, overlapping and fragmented.

To realize this level of problem identification, the focus will be put on the following activities:

- Identifying concerns.
- Breaking down concerns into manageable components.
- Prioritizing concerns.
- Solving concerns.

In the following, the four activities are discussed in detail and applied to the PDMS expert system knowledge-base.

3.2 Identifying Concerns

A concern is defined as a situation or a case requiring action that can be addressed by the evaluation. When trying to correct a knowledge-base that was built by different field experts, we can not afford to jump quickly to conclusions. Evaluators should not look at a problem with the objective to solve it, at least, not at this point. Thus, several activities should be realized before reaching the step of solving the existing problems. First, the concerns have to be identified and clearly stated. Secondly, the stated concerns have to be well defined in order to have the same meaning in the mind of the evaluators. Some concerns are clearly identified and simply addressed, others are present but not easily pinpointed.

To methodically address and clearly identify the concerns, the search for a solution has to be broken down into four different activities:

- To list all deviations from the primary goal, taking into consideration the threats and the opportunities.
- To review the progress during the evaluation process creating a feedback to adjust the evaluation and keeping the objectives in mind.
- To foresee potential problems and surprises.
- To search for improvements.

At the beginning, one wants to go through the first two steps which are to list the deviations and to review the progress in the evaluation process. But during the knowledge enhancement process, the focus is inevitably put on the third and forth steps.

The first step is to list the goals that have been set at the design stage, and which the product has to meet. One must then verify if these goals are met.

Listing the deviations between the initially pursued goals and the tested product is not an obvious task that can be easily executed. To help perform this step of problem identification at an acceptable level, the activity can be broken down into the following tasks:

- To list the goals the product has to meet, which have been set during the high level design.
- To list all deviations from the design objectives.
- To list persistent problems.
- To list all issues currently under investigation that will be solved in a near future (this will help eliminate already identified problems)
- To list all reasons used during the problem identification.
- To list all decisions that require action.

In following this method, a step is made toward an eventual identification of concerns. All known problems are listed with a clear and consistent understanding of the problems in the evaluators mind.

At this point, no conclusion can be drawn. Each concern has to be examined to determine whether it is an isolated problem or a composite problem that need further breaking down.

3.3 Breaking Down Concerns

It is difficult and confusing to deal with a combination of concerns that appear as a unique case. Even if a particular problem is understood, it does not mean that it can be solved. An easily identifiable problem can be caused by the co-existence of different pieces of knowledge interacting together. Thus, if one tries to solve what appears to be the cause and which actually is a multitude of problems, it will be patching a particular case and introducing distortion in the knowledge-base. Therefore, one should assume that all identified issues are complex in order to ensure that the information gathering process is complete and ready for the concerns evaluation. At this stage, all cases that require action or attention are identified. Now, the objective is to break apart cases that include two or more components. The following questions make the task easier when addressed:

- Is one or several issues addressed?
- Is there a clear understanding of the concern and the reason to address it?
- Is there an evidence that this is a true concern?
- What is happening (in opposition to what is supposed to happen)?
- What observed result indicates the defective functioning of the system?
- How is the error affecting the outcome?
- Does one action resolve the issue?

Eventhough, the previous questions seem to overlap, they represent different angles for viewing a concern. when all the questions are taken together, they enable the "evaluator-developer" to shift the emphasis from opinions to information gathering, and thus, to elicit data ready for the evaluation process.

Moreover, at this step, the evaluator should refer to all the resources that allowed acquiring the encoded knowledge.

The purpose of the "breaking down concerns" step is to ensure that all known primitive concerns are gathered. The role of primitive concerns identification is increasingly important with the introduction of new concepts such as knowledge

framework and knowledge reuse. These new techniques encourage the knowledge interaction which, in turn, make the different concerns of a problem transparent to the evaluator. The virtue that these new techniques bring to the current implementations are considerable. But, the complexity of the evaluation is far greater. There is a need to impose on the evaluator predefined rules that will ensure a good result leading to a good design and a good product.

3.4 Setting Priorities

Now that the concerns are broken down into manageable components, they must be prioritized. The step of setting priorities consists of listing the issues and concerns in an ordered way according to their importance which will lead to the definition of the relative importance of each issue. What does importance mean? As the importance is very subjective, it becomes primordial to establish a practical and systematic process for determining importance. In order to achieve a uniformity in the definition of the importance of each issue, every concern has to be considered in terms of the three variables listed below:

- Seriousness: how serious is the current impact of the issue on the product goal.
- Urgency: how time critical is the issue. (hierarchy depending on other issues)
- Probable growth: the estimate of the probable growing importance of such an issue.

The importance of each variable is tabulated for each issue. In this way, concerns, now manageable, can be tabulated in an ordered way according to their priorities.

At this point, any evaluator can examine the high priority issues and neglect those with a low ranking on the three variables. This does not mean that the low rank issues are to be eliminated but they should be postponed until they top the list of concerns.

To illustrate the use of the three variables responsible for prioritizing the identified concerns, we propose to consider the example of a medical expert system presenting two

problems. The system is assigning the value of the blood pressure to the heart rate, and the system is using the wrong color map for the display of the patient condition. The seriousness of both problems are equal, in that they both have a direct impact on the product goal. Assuming that the user can inspect the condition of a patient in a written format, the urgency of the first problem becomes greater then the second, as a work around the color problem exists. Finally, the growth of the first problem is again greater than the second, as it will affect the decision making of the entire expert system. Tabulating the results, as shown in Table 3 will help prioritizing the two problems at hand.

Table 3 Prioritizing example

	Vital Sign Problem	Color Problem
Seriousness	High	High
Urgency	High	Low
Probable Growth	High	Low
Priority	1	2

Once the three previous steps completed, a number of issues are thus collected. Now, the next and final step is to solve each individual problem. At this point the evaluators should decide on one of the following actions to take:

- If the root cause of the problem is understood, and there is an evident solution, then an action has to be taken to solve the problem.
- If the root cause of the problem is understood, but different controversial solutions exist, then a decision should be made on which solution to implement.
- If the issue is still not completely understood, then the process has to be repeated.

At this point, all issues known are identified and presented in an orderly and manageable way. They can be directly worked on with the objective to solve them. The process described above does not achieve the expected results if it is not repeated several times. It relies on the consistency of the information among the evaluators, of the

information sharing during the evaluation and on the clarification of the problems. Once all these major points covered, correcting the defects of the problem becomes possible for the designer without requiring the understanding of the encrypted knowledge that exists in the expert system.

A graphical representation of the process is illustrate in Figure 5.

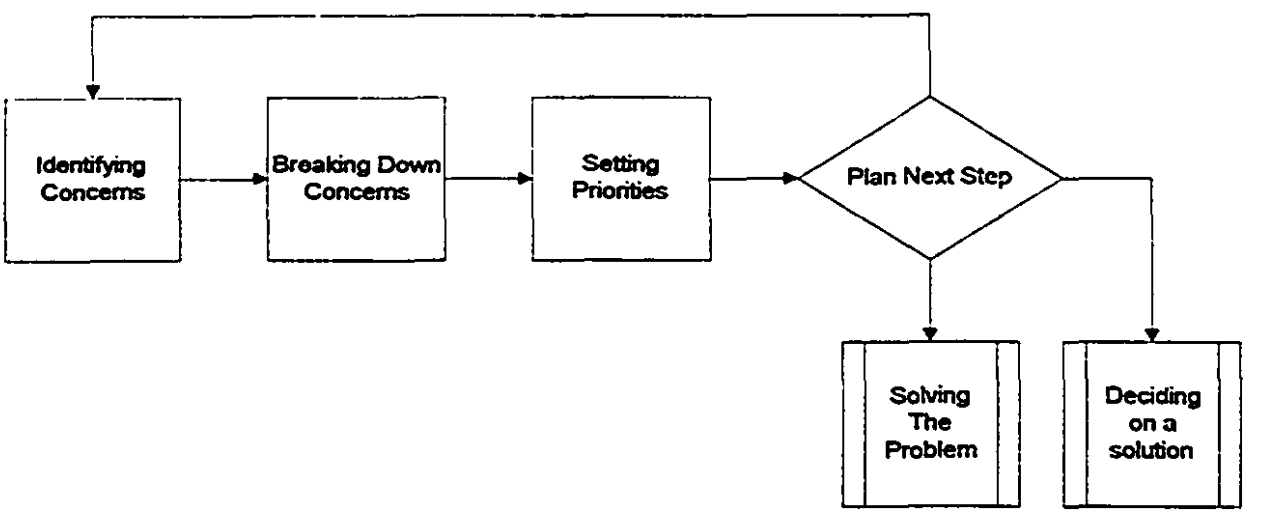

Figure 5 Process overview

3.5 Evaluation Process Visualization

The evaluation process described above enables the people responsible for the evaluation of the PDMS expert system knowledge-base to identify the problems and to develop action plans to solve them in a systematic way. The process is composed of different steps that have to be completed sequentially. In this section, a visualization of the process is presented to the evaluator in order to show the benefit of such a process. This representation of all the steps relies on the fact that the process works best when all the steps are presented to the evaluators at the same time. Moreover, it can facilitate the task that is required from the evaluator.

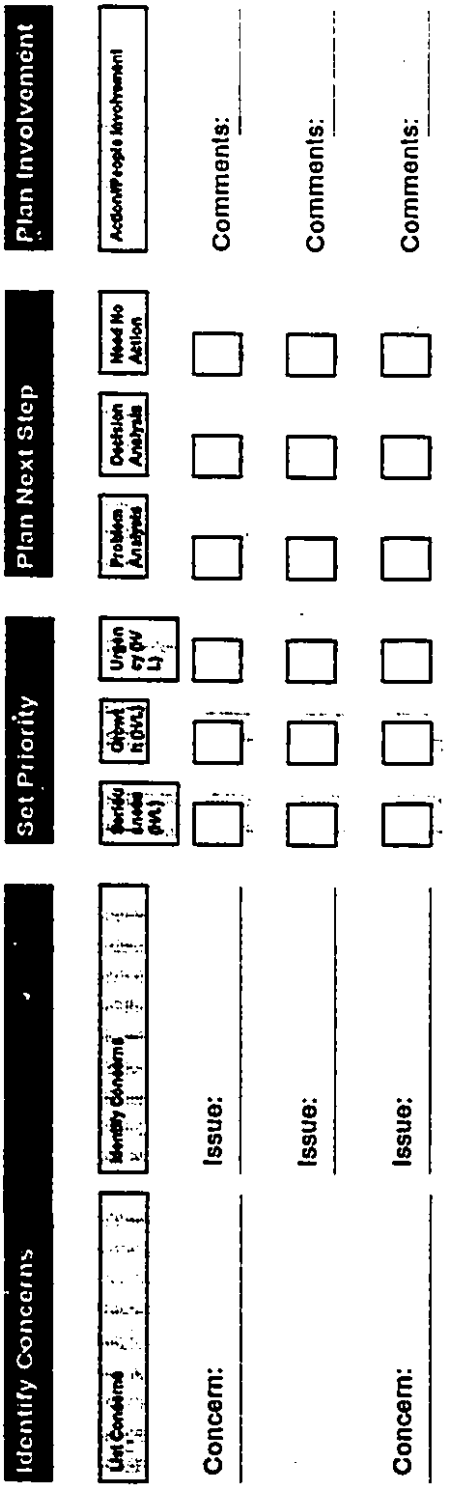

Figure 6 Evaluation sheet

The best way to allow the user of the process to have a snap shot of all the concerns, issues, priorities and plans is to present all the information in a tabular fashion as in Figure 6. A reference card is shown in Figure 7 to facilitate and accelerate the evaluator's work.

3.6 The Evaluator Attitude

At this stage, the evaluator has to adopt a systematic approach to the product evaluation, whether she/he is at the same time the designer, a contributor to the development of the product, a contributor to the knowledge-base or a field expert. The person involved in the evaluation should study the case without trying to analyze the problem or to come up with an action to take. The evaluator should focus exclusively on appraising the case. The evaluation schema have all the problem solving and decision making for the product evaluation. The evaluator should only answer the questions (in the case of a questionnaire). If this attitude is not respected, it only introduces uncertainty, confusion and overlapping possibilities to every situation, which will yield inefficacy and incorrect results.

3.7 A Two-Step Evaluation

The evaluation process must be an integral part of the life cycle of a product. Nevertheless, in many cases of software development, the evaluation process is executed in a posteriori fashion. This is the case with the PDMS. To get around this problem, the evaluation process is broken into two distinct steps: the *initial evaluation* and the *progressive evaluation* of the system. Each of the two evaluations is explained in the following section.

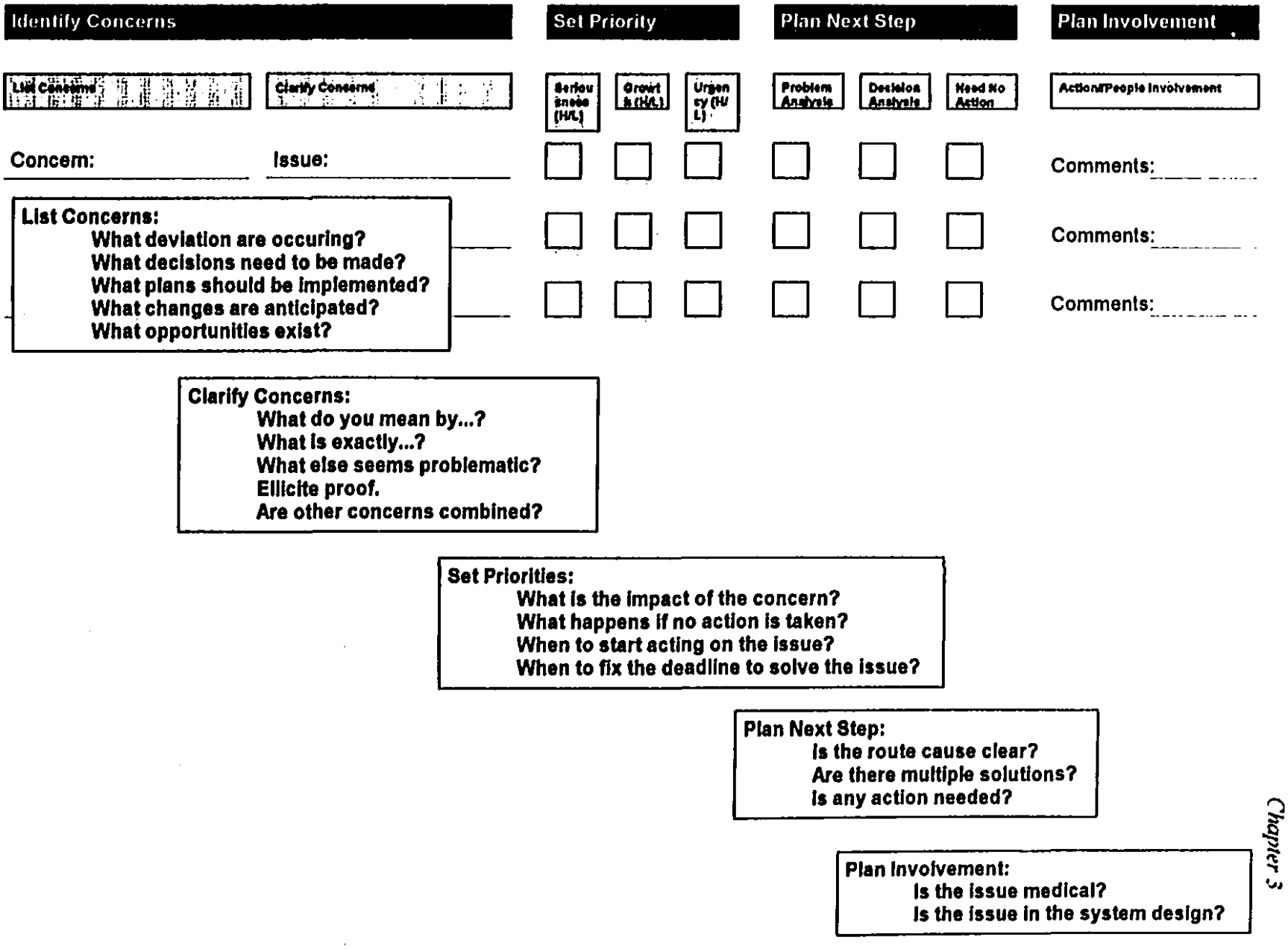

Figure 7 Evaluation reference sheet

3.7.1 Initial evaluation

The initial evaluation is used as a starting point for the overall system evaluation. As the expected number of errors present in the system can not be initially predicted, the worst case is taken into consideration to establish the base-line level of errors in the system. By assuming that there is a large amount of problems present in the knowledge-base, the objective of the evaluation process is to ensure the reliability of the system before it can be put on the field. Unfortunately, this objective is difficult to quantify. The engineer can not ensure an error-free functioning, and the medical staff can not guarantee a correct diagnosis for all conditions. In order to overcome these theoretical and empirical limitations, an initial testing was implemented to detect the software bugs and the knowledge-base problems.

The initial evaluation has to bring people's confidence in the product to an acceptable level. It also has to be short enough in order not to make people loose interest in the product. In the PDMS case, an adequate and feasible objective is to randomly pick ten cases for every diagnosis the PDMS expert system generates and put them on the initial evaluation. This would lead to 120 different test cases, considering that one should not over look the case when no alarms are generated.

When choosing the cases, one should try to obtain the maximum number of variety in situations that leads to one specific diagnosis. This will allow a better visualization of the knowledge-base. The following parameters should be considered when trying to pick the cases: the patient condition alarm, the predicted patient condition alarm, the dynamics of the vital signs, the age and sex of the patient, the duration of the alarm. An example of a good choice would be to have in one case a highly fluctuating alarm and in the other a steady alarm.

Once the cases are chosen, the evaluators (mainly medical experts) should make a judgment of the correctness of the diagnosis based on the data presented to the expert system (the three main vital signs) and based on the data available. At this point, apparent

problems are kept for further discussion with the knowledge engineering staff. This will result in a problem identification process leading to a knowledge enhancement.

The initial evaluation covers a wide area of the knowledge-base and induces a confidence towards the system in the user's mind. It also allows the software designer to perform a field test on the end-product.

If the evaluators feel the need for further examination, it should be performed. These extra evaluations should be more targeted on the weak areas of the knowledge-base.

3.7.2 Progressive evaluation

The assumptions and the objective underlying the progressive evaluation are very different from those formulated in the initial evaluation. The purpose of the progressive evaluation is to enhance and improve the knowledge-base in order to meet the user's increasing expectations from the system. It assumes that the system is used on a continuous basis and the user has confidence in the system.

The operation of the progressive evaluation is problem driven. At this point, the system should produce adequate performance. There is no need to collect information for the evaluation anymore. The process is to record every mis-diagnosis generated by the system. The following information has to be present in order to allow a good case evaluation:

- Patient's name.
- The time the problem occurred.
- Comments on the situation.
- Description of what is believed to be the reason for the firing of a problem report.

The method presented helps the evaluators make sense of concerns that actually are unruly collections of concerns, each with its own requirements. The method will help

cut down the amount of time and energy wasted on misunderstanding and misuse of information. It will help generate production actions by setting priorities. If respected, the objective of the technique is to make the appropriate actions in order to resolve concerns.

4. Evaluation Results

The implementation of the evaluation process of the PDMS medical expert system knowledge-base involves the setting of a well defined methodological process that the user can follow. Different elements are added to the questionnaire described previously in order to help trace, isolate and correct any problem. Before any evaluation is performed, a number of steps need to be performed in order to be able to execute evaluation and to make the process feasible and friendly.

When trying to evaluate the PDMS medical knowledge-base, numerous issues play a decisive role in the realization of the task. Data collection is the basic element needed to perform the evaluation. Other issues, such as, data visualization, case identification impact on the presentation of the document to the evaluator and, therefore, on the effort the evaluator will invest in the evaluation. Each of these issues plays a decisive role on the success of the evaluation process.

In this chapter, the preparation of the evaluation is broken down into five steps. The first step restates the evaluation topic to reinforce the core subject of the study. The data collection, a new functionality added to the PDMS in order to allow the reading and storing of the data needed for the evaluation, is presented. The case identification and presentation, needed to keep the evaluation user friendly, is described. And, the last step gives guidelines on planning people involvement.

4.1 Evaluation Scope

This section restates the topic of the evaluation with the intention of imposing defined boundaries. It is very easy, in a topic such as knowledge-base evaluation, to see

the group of evaluators drifting from the real subject or being confused about the real topic of the evaluation. Therefore this section sets the scope to the evaluation.

It is very important to clarify to the evaluator what is being evaluated in the PDMS. The PDMS is a product that provides various functionalities and has a large amount of lines of code. The evaluator should be focused on the object of the evaluation and not on some other feature present in the PDMS. A simple example would be an evaluator worried about the correctness of the vital signs reading. This is a valid concern, but not during the evaluation of the expert system knowledge-base.

The goal of the PDMS expert system is the *patient's condition representation and correct diagnostics*. To meet this goal, the following concerns are identified:

- Acceptable diagnosis.
- Software robustness.
- Real-time capability.
- User-friendly interface.

The scope of this evaluation is limited to the verification and validation of the knowledge-base of the EMS. Therefore only the concern of providing *acceptable diagnosis* is considered in the evaluation. The other concerns such as system performance have been evaluated and are documented in other studies [Lam, 1993].

4.2 Required Feature

In order to be able to perform the evaluation, a number of data collections should be performed from various parts of the PDMS software. A new feature was introduced in the PDMS product to collect the data needed. This added functionality is invisible to the user, and does not introduce any new module.

The data collection is performed in the PDMS by means of centrally located and centrally generated logs. The log system is used by the PDMS software to record all

significant data generated for each registered patient. The log system creates a report for every registered patient containing the following information:

4.2.1 Header information:

- Bed number.
- Patient name.
- Patient birth date.

4.2.2 Running log:

- Patient recorded vital signs.
- The three main vital signs unprocessed: arterial blood pressure, heart rate, central venous pressure.
- The classified level and next level of the three main vital signs in raw and processed value.
- The vital sign processed.
- The certainty indices.
- The predicted certainty indices.
- The current and predicted alarms and their certainty indices.
- Current date and time.
- Time in seconds with respect to 01/01/1980.

Figure 8 shows an example of a log file. The patient recorded vital signs field show

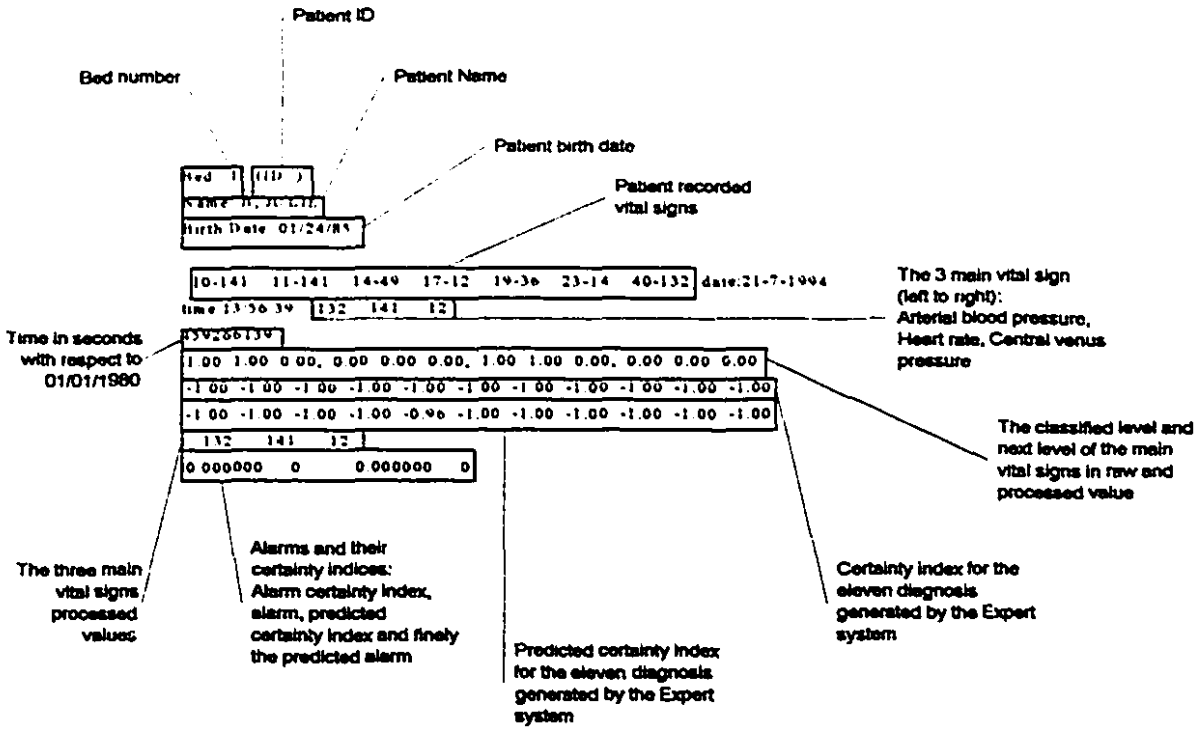

Figure 8 Log file description.

a serie

of two numbers, number1-number2. Number1 represents the vital sign code in the Data Link Controller, number2 represents the value of the vital sign. The vital sign classified levels and next level are represented in the following order: current level of blood pressure (BP) raw data, heart rate (HR) raw data, central venus pressure (CVP) raw data, BP processed data, HR processed data, CVP processed data, next levels of BP processed data, HR processed data, and CVP processed data, BP raw data, HR raw data, CVP raw data. The certainty indices and predicted certainty indices fields represent the confidence of a diagnosis, listed in the following order: agitation, bradyarrhythmia, CNS-Drugs, CNS-ICP, primary hypertension, hypervolemia, hypovolemia, pump failure, systemic shock, tachyarhhythmia, tamponade.

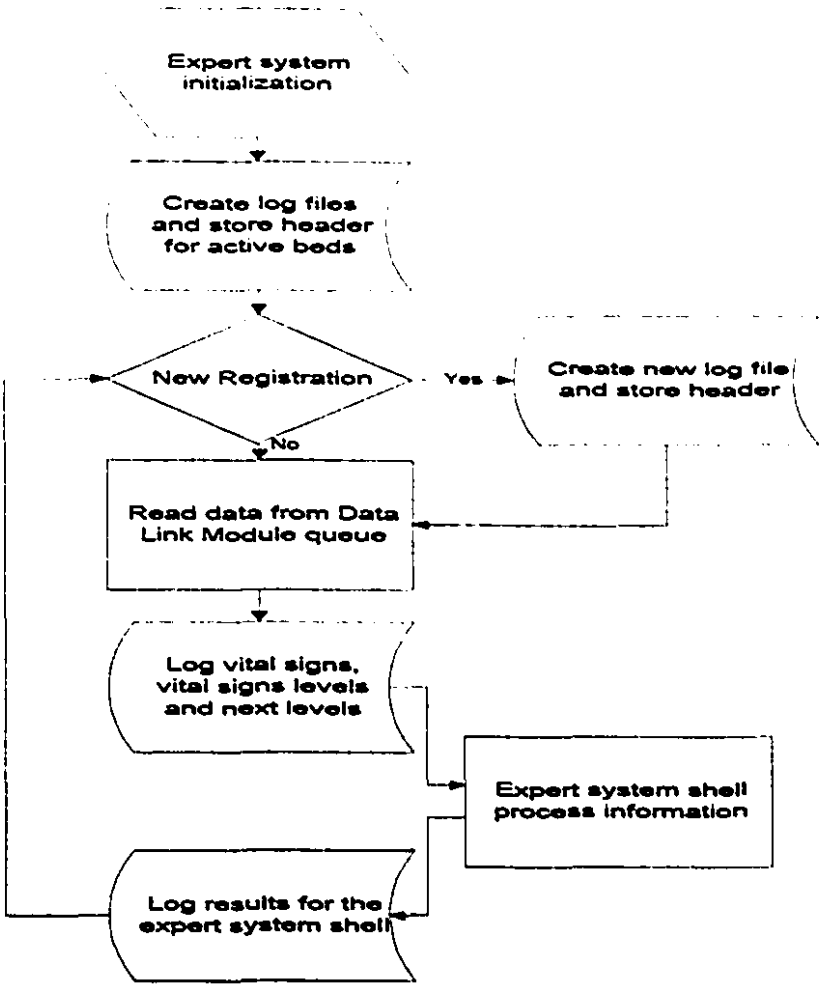

Figure 9 Introduction of the log system in the expert system.

The logging system software is centralized in the expert system module. It involves a series of software hooks placed in targeted places having minimum impact on the real-time operation of the expert system.

Bed files are created and information header is written during data initialization or registration time depending on the time the patient is registered into a bed. The vital signs levels and next levels are written as they are sent to the expert system shell. The rest of the logs are written to the log file after being calculated by the EMS. Figure 9 shows how the different logging events are imbedded in the expert system software.

The log files are stored in flat text files to simplify the task of reading them as well as to allow simple porting to other systems for off-line analysis.

4.3 Test Case Selection

Various methods exist for selecting test cases. Frequently these are based on a statistical approach. For simplicity, this was not used in the PDMS evaluation. The reason is to minimize the effort required in collecting the evaluation data. This section present the two phases of the PDMS evaluation: the initial and the progressive evaluation.

To serve the purpose of the knowledge-base evaluation process, two different methods of test case selection are used. The first case selection method is for the initial knowledge-base testing. A random selection of test cases is chosen from the logged expert system patient condition evaluations. Ten cases were sought for each of the eleven possible medical conditions (CI) to serve for the pilot testing. The selection of the cases is based on the alarms and CI's generated by the expert system. A case is identified as a sixty minute recordings centered on the expert system generating an alarm for a wanted medical condition. From the patient data available at the time of this initial evaluation, recording of all possible patient conditions could not be located and approximately 30 interesting data sets were selected for analysis.

In the second phase of the PDMS evaluation, we will assume that we are dealing with an enhanced version of the initial knowledge-base. The assumption is that most of the major faults present in the knowledge-base were corrected in the initial case identification. Now that a clean knowledge-base is in the system, the user performs the case identification on a per error basis. In the future, the arrival rate of problems is expected to be low, allowing the user/evaluator to progressively enhance the knowledge-base, resulting in an adaptive and progressive system. If the assumption of the second phase of the evaluation fails, the system is still in an unstable situation; in this case, jumping to the second phase will result in an ineffective evaluation.

4.4 Case Presentation

This section discusses the data presentation and data analysis that are performed for every case. An overview of some results are presented.

Every case presented to the evaluator should contain all relevant information needed to perform the analysis and evaluation of the covered part of the system. In order to keep the case studies simple, the evaluator is presented with a sixty minutes time segment of the patient data. The following data is presented in order to allow a global picture of the state of patient and the system:

- The minute values of the raw or original measurements of the heart rate, the blood pressure and the central venus pressure. This data is calculated by averaging the second date collected from the HP Carenet.
- The processed minute values of the heart rate, blood pressure and central venus pressure. The EMS system relies heavily on this data to generate the diagnosis. It shows the efficiency of the processed data to track, smooth and eliminate the spikes in the original data.
- The actual patient condition.
- The predicted patient condition.
- The certainty indices of the generated patient condition diagnoses.

Figure 10 to Figure 16 show an example of a set of data trends used in evaluating the performance of the EMS for a one year old male. Figure 11 to 13 represent the different actual conditions present in this case. Figure 14 to 16 represent the different processed conditions present in this case.

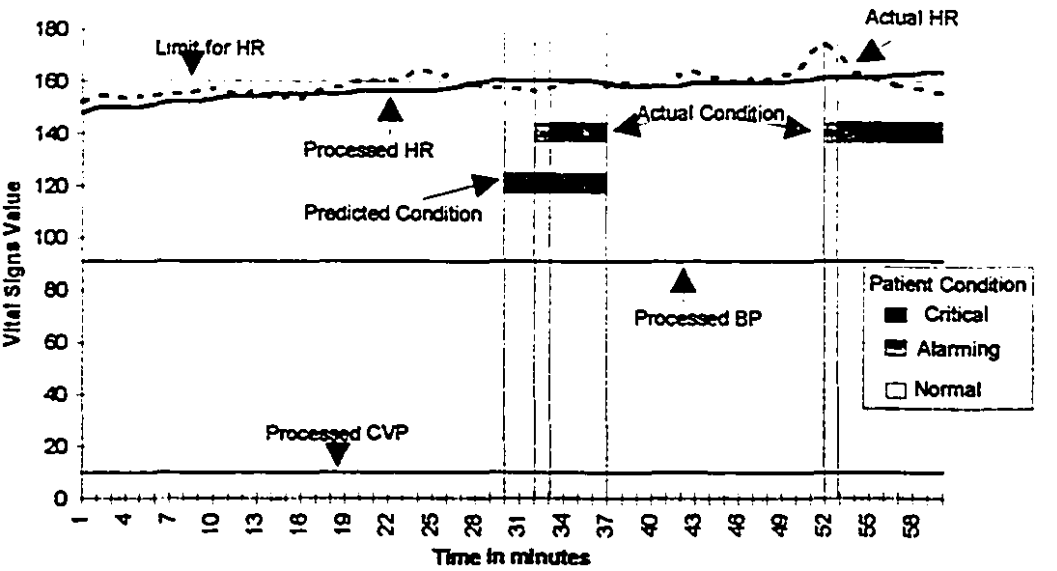

Figure 10 Vital sign and patient condition trends

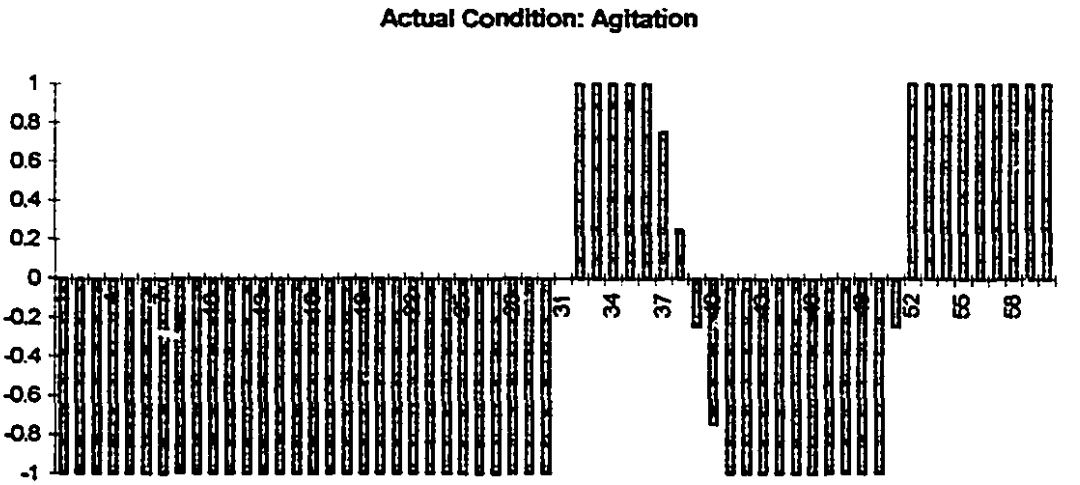

Figure 11 Actual agitation condition

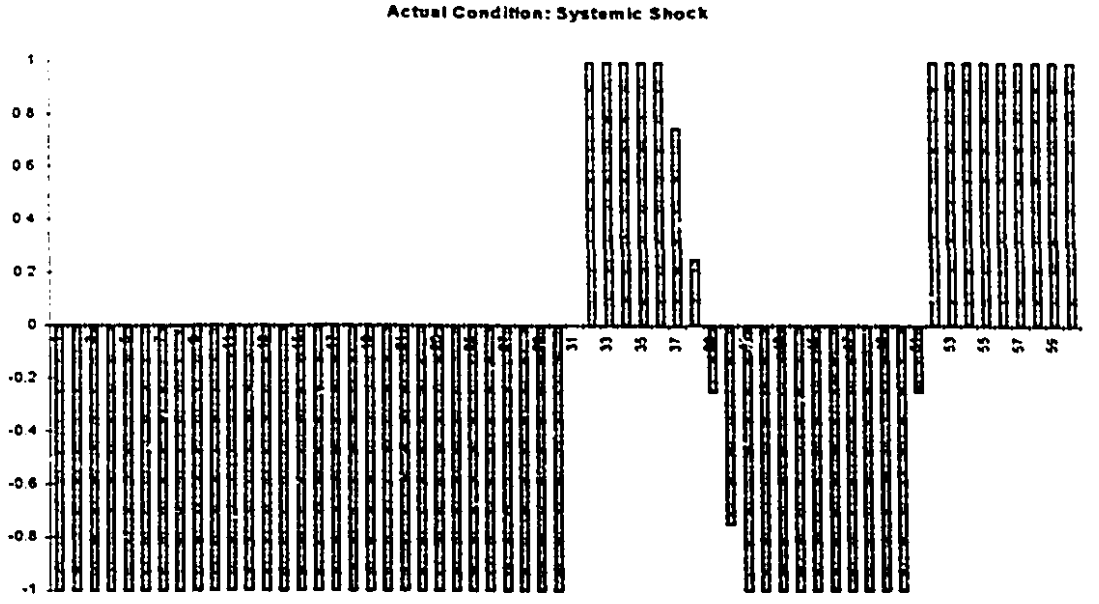

Figure 12 Actual systemic shock condition

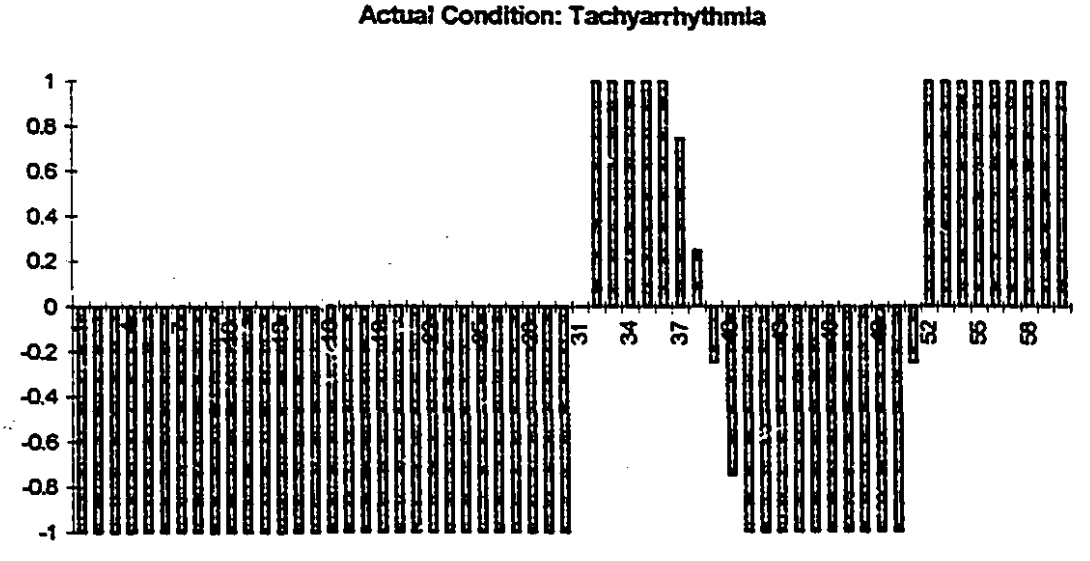

Figure 13 Actual tachyarrhythmia condition

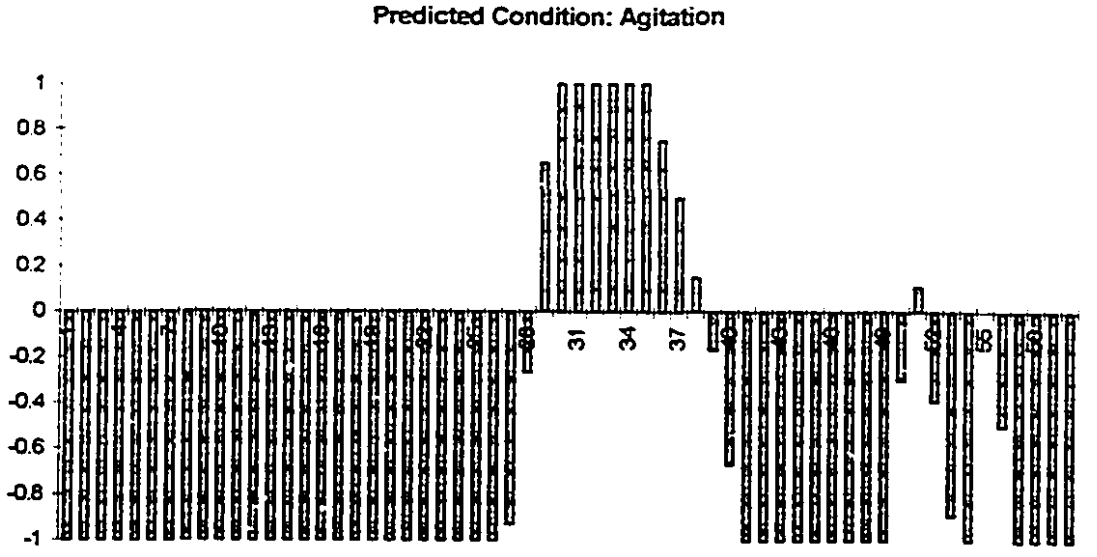

Figure 14 Predicted agitation condition

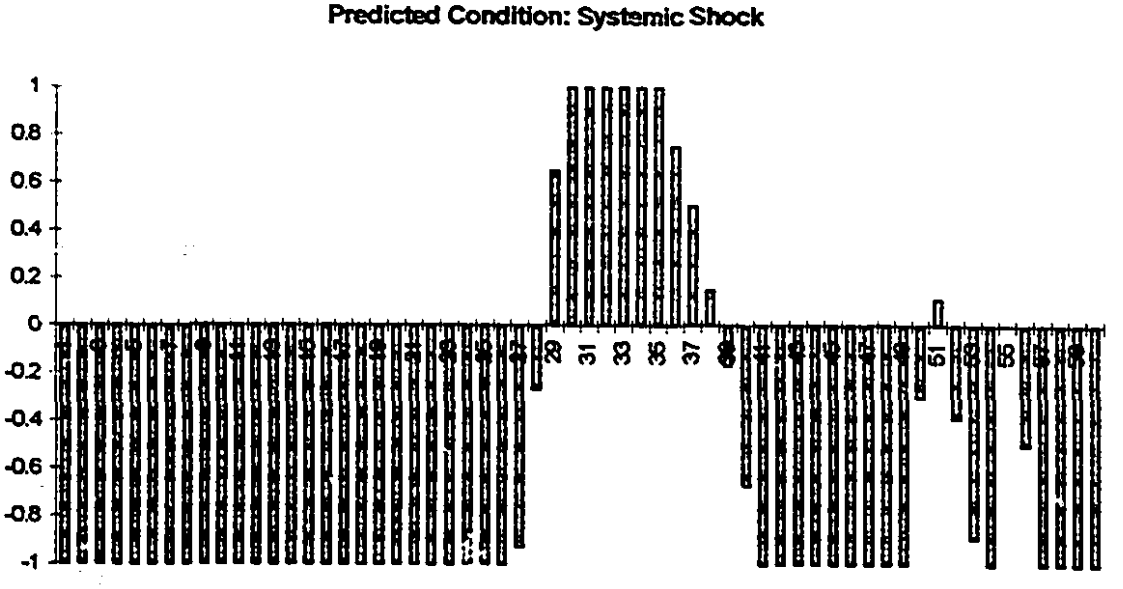

Figure 15 Predicted systemic shock condition

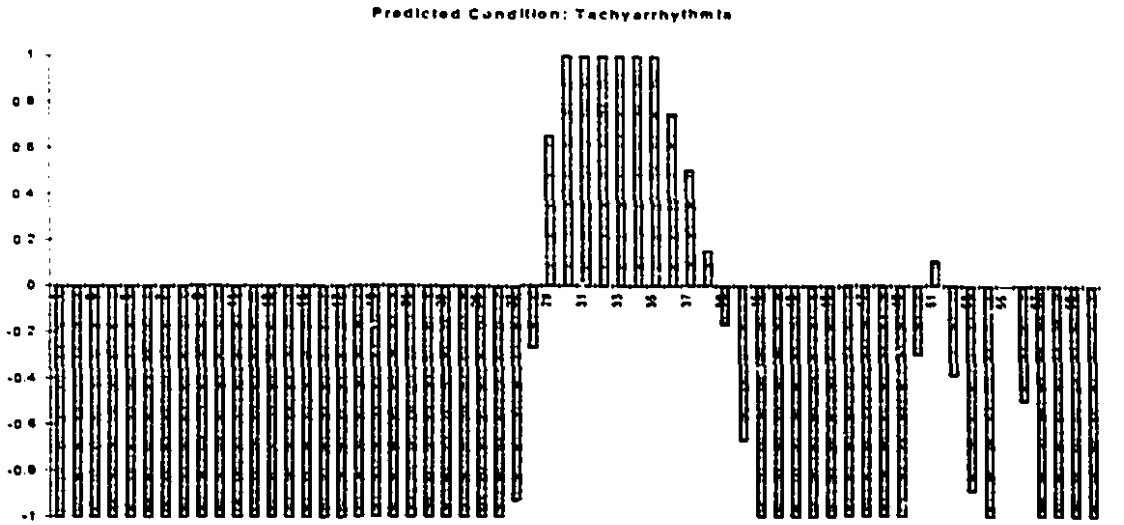

Figure 16 Predicted tachyarrithmia condition

4.4.1 Data Analysis

The data presented in Figure 10 through Figure 16 will now be explained. This example illustrates a condition diagnosis transition from normal to alarming and then to critical. This study will focus on the overall patient condition. The objective of this section is simply to give an understanding of the system dynamics and the reason behind these transitions. It is also used to explain the different processing invisible to the user. The medical soundness of the CI's will be discussed later.

The different parameters mentioned previously are labeled on the graph. Notice that only one parameter was collected in this example. The system allocates a normal default value to the other two vital signs. The default values are chosen according to the

age of the patient. The actual and predicted condition is illustrated using different shadings on the graph.

Note that only the medical condition that are "active" (i.e. have a significant CI) are shown in this example

4.4.2 Data Analysis

The graph of Figure 10 illustrates the heart rate transitioning from a normal state to an alarming state. Different thresholds are set for every vital sign depending on the age of the patient. In this case where the patient is one year old, the thresholds settings are shown in Table 4.

Table 4 Vital signs threshold values

	Critically High	Alarmingly High	Alarmingly Low	Critically Low
Blood Pressure	150	110	72	42
Heart Rate	200	160	80	60
Central Venus Pressure	20	15	5	3

At t = 32 minutes, the calculated overall patient condition enters the alarming condition by crossing the yellow threshold of 0.5. At t = 32 minutes, this condition becomes critical when it crosses the red threshold value of 0.75.

4.4.2.1 Processed Data Validity

This section analyzes the performance of the processed data and makes an attempt to evaluate it according to the original objectives set for the EMS system.

The objective set for the processed data is to provide the system with the following features:

- Smooth the data received from the network
- Provide a good tracking of the original data
- Eliminate the inherent noise
- Avoid the spikes present in the raw data

Figure 10 shows that the smoothing and tracking of the raw data is effective. The filtering provides a conservative data that is still representative of the original data. The spike elimination can not be observed in this case but was proven several times in other recordings carried out during the field tests.

4.4.2.2 Actual Condition Validity

It is shown on the graph of Figure 10 that the diagnosis of the system starts by generating a yellow condition as the system is not positive that the condition is valid. As the patient vital signs persist in a definite non-normal region, the system repeatedly concludes that the medical condition identified is valid and generates the red condition. The red condition is the result of the accumulation of consistent occurrences of a medical condition and vital signs residing in an alarming or critical region.

4.4.2.3 Predicted Condition Validity

The calculation of the predicted condition involves more parameters than the calculation of the actual condition. The predictions require the following additional variables to achieve the desired response:

- Time difference between readings
- Direction of the change of the vital signs (going towards or away the limit allowed)
- How fast the vital sign is approaching the limit allowed
- Further consideration of the raw data

Examining the prediction is less intuitive than examining the actual condition. Although a condition can go alarming for an period of time, it is valid for the system to predict that the patient is going towards a normal condition for different reasons. The following illustrate some of the reasons of the case described previously:

- The raw data experiences some fluctuation towards an acceptable value
- The time difference between two data readings is too large to allow an acceptable prediction
- Although some vital signs can be in the alarming region, the rate of change that the vital sign experienced to reach the limit is too small (note that if the vital sign persists for a large period of time in an alarming or critical region, the actual value of the vital sign will have a greater weight then the rate of change and will push the system to predict an alarm).

Still, the most important factor in determining the prediction is the rate of change and the proximity of the data to one of its allowed limits.

The previous graphs illustrate the aforementioned behaviors. For the case of figure 10, in the period of time between 29 and 38, the system predicts an alarm due to the steep fluctuation of the heart rate towards its limit of 160. In the case of figure 10, for the period of time greater than 51, the system predicts that the medical condition will

disappear as the fluctuation is not drastic and the raw data is dropping towards the normal value of the heart rate.

4.4.2.4 Analysis of Patient Condition Results

The EMS system was installed for field testing at the Montreal Children's Hospital in May 1993. The initial goal of field testing was to assess the robustness of the system under real operating conditions. Then test results were collected to evaluate the EMS and determine some improvements that would help the current system to perform at a better level.

Table 5 Modified medical rules

Heart Rate	-	-	-	-	-	-	-	-	-	=	=	=	=	=	=	=	=	+	+	+	+	+	+	+	+	+
Blood Pressure	-	-	-	=	=	=	+	+	+	-	-	-	=	=	=	+	+	+	-	-	-	=	=	=	+	+	+
Central Venus Pressure	-	=	+	-	=	+	-	=	+	-	=	+	-	=	+	-	=	+	-	=	+	-	=	+	-	=	+
Hypovolemia																			x			x					
Hypervolemia						x			x												x			x			x
Bradyarrhythmia	x	x	x	x	x	x																					
Tachyarythmia																			x	x	x	x	x	x			
Tamponade																					x			x			
Pump Failure										x										x	x			x			
CNS-ICP				x	x	x	x	x	x																z	z	z
CNS-Drugs	x	x	x	x	x	x				x	x	x															
Primary Hypertension							x	x	x								x	x								x	
Systemic Shock											x	x							x	x	x	x	x				
Agitation																						x	x	x	x	x	x
Normal													x														

About 25 megabytes of vital sign data was recorded with the PDMS and the EMS over the period of 3 months. This period of time did not allow the collection of all the varieties of patient conditions desired, but allowed to examine real patient data samples. These data files were migrated to a SUN UNIX workstation . Using XWV, a viewing and plotting tool, these data files were reviewed and approximately 30 data sets were

extracted such that all three vital signs were captured and some interesting changes were visible in the bed conditions and associated medical diagnoses. Using MATLAB (a product of MathWorks company) these files were printed as outlined in section 4.4 for study by the medical experts. These studies resulted in a better understanding of how the EMS performed and suggested modifications to the rules of the knowledge-base. The primary and fundamental basis of the EMS is the medical knowledge selected and encoded in the expert system rules. As a result of this evaluation, the medical diagnostic rules have been modified to improve the performance of the EMS, especially its selectivity and its predictive capability. These modified rules are presented in Table 5. The modified system was then implemented and subsequently re-assessed [Abu-Shihab, 1996].

4.4.2.5 Future Work

Another important aspect is the dynamic response of the EMS which is related to the various weighting factors and filtering coefficients used in the design. The PDMS is currently being extended to include the OS/2 Database Manager for filing [Saab, 95]. With the eventual integration of these database facilities in all the PDMS modules, it will become much easier to select the appropriate data sets for more elaborate evaluation studies. The medical diagnostic rules could then be elaborated to include medications and other patient conditions.

Another suggested improvement is the implementation of the rule-based system using a more efficient environment such as CLIPS rather than the present NEXPERT environment.

An important practical enhancement to the PDMS system should include automatic reboot capabilities and a battery backup system to deal with the inevitable power transfers or outages. Database recovery procedures should also be included.

The original architecture shown in Chapter 2 has limitations and an obvious enhancement is to take advantage of current distributed computing technologies and

client/server architectures. Today's technology allows the system to distribute its functionality over a network of computers. The introduction of a distributed computing environment, already supported by IBM for the OS/2 operating system, will enhance the processing power available to support a greater number of patients as well as the continual addition of the new software modules being developed.

4.4.3 Planning People Involvement.

Planning people involvement is one of the most important steps of the evaluation. It dictates the success or the failure of such a costly task. This section describes the people needed to ensure a successful evaluation.

To correctly recognize all the concerns, the people who participated in the development of the product should participate to the evaluation activity:

- The product architect: to input concerns that emerged during the product design and to correct other concerns if they were not accounted for in the original concept.
- The product developers: again, the concerns of the developers help ensure that no major design faults are present. They also introduce a feedback in the design loop as they know how closely the product related to the design.
- The field expert: in the PDMS case, the doctor and nurse play that major role. The field expert is a user highly knowledgeable in the field.

All findings or suggestions must be clearly stated and the information must be available to all the evaluation participants. The identified people must systematically cycle through the questions of the concern identification until everybody agrees on the confidence in the product.

It is frequently very difficult to ensure the presence of the original architects, designers and field experts in a long-term project. However, this ideal situation is more likely to be met in a short-term project.

5. Conclusion

The system presented has been installed at the Montreal Children's Hospital ICU. The field tests have helped correct and improve the original system. As it stands, the PDMS, is operational as a research tool. The on-going medical evaluation process is continuing and more comprehensive studies are envisaged when the database support is fully installed. This will permit future evaluators to fine tune the medical knowledge-base of the PDMS, and increase the confidence in the product.

This thesis presented the evaluation of the knowledge-base of the PDMS by presenting the different current expert system evaluation process currently available. The PDMS system is then presented describing the hardware responsible for collecting the patient's vital signs and the software modules responsible for data communication, patient registration, database management, vital signs monitoring, fluid balance monitoring, nurse workload management and patient diagnosis generation and prediction. The evaluation and testing of the medical expert system is then introduced.

It is still impossible to predict the quality of the knowledge-base. This thesis attempts to introduce an approach to enhance and effectively troubleshoot an expert system knowledge-base without having to re-engineering it. The first steps achieved in the initial evaluation allowed the correction of some software bugs and showed some weaknesses present in the knowledge-base and these were subsequently corrected. Unfortunately, it is difficult to achieve the complete evaluation process with ease due to the nature of medicine. Cases are not generated on user's request but depend upon the

availability of the patients present in the ICU. This makes the comprehensive evaluation of the EMS tedious ongoing process.

Glossary

BP	Blood Pressure, an input parameter of the Expert Monitoring System
CI	Certainty Index, a number used to reflect the accumulated number of times particular patient condition is found
CPU	Central Processing Unit
CVP	Central Vencus Pressure, an input parameter of the Expert Monitoring System
CVS	Cardiovascular System
DLC	Data Link Controller, a software module of the Patient Data Management System
EMS	Expert Monitoring System, a real-time monitoring and warning expert system developped at McGill Universityfor the Patient Data Management System of the Intensive Care Unit of the Montreal Children's Hospital
FIR	Frequency Impulse Response, one of the filters used in the filter module of the Expert Monitoring System
HR	Heart Rate, an input parameter of the Expert Monitoring System

ICU	Intensive Care Unit
MWCIN	A rule-based expert system developed at Stanford University to determine diagnosis and recommend treatment for infectious diseases
PDMS	Patient Data Management System, a personal computer-based information management system developed at McGill University for handling patient data in the Intensive Care Unit of the Montreal Children's Hospital
PM	Presentation Manager
PRN	Progressive Research in Nursing
RAM	Random Access Memory

References

[Abu-Shihab, 1996] O. Abu-Shihab, "Implementing and Evaluating a Vital Sign Monitoring System in an ICU" Master's thesis, McGill University, Montreal, Quebec, Canada, March 1996.

[Bailey, 1983] J.E. Bailey and S.W. Pearson, "Development of a tool for measuring and analyzing computer user satisfaction," Management Science, 29(5), p. 530-545, 1983.

[Baroudi, 1988] J.J. Baroudi and W.J. Orlinkowski, "A Short-Form Measure of User Information Satisfaction: Apsychometric Evaluation and Notes on Use," Journal of Management Information Systems, 4 (4), p. 44-59, Spring 1988.

[Boehm, 1976] B.W. Boehm, J.R. Brown and M. Lipow, "Quantitative Evaluation of Software Quality", Proceeding of the Second International Conference on Software Engineering, p. 592-605, 1976.

[Buchanan, 1984] B.G. Buchanan and E.H. Shortliffe, "Rule-Based Expert Systems: The MYCIN Experiment of the Stanford Heuristic Programming Project", Addison-Wesley, Reading, Massachusetts, 1984.

[Doll, 1988] W.J. Doll and G. Torkzadeh, "The Measurement of End-User Computing Satisfaction", MIS Quarterly, 12(2), p. 259-274, June 1988.

[Fox, 1990] M.S. Fox, "AI and Expert System Myths, Legends and Facts", IEEE Expert, 5(1), p. 8-20, February 1990.

[Fumai, 1990] N.Fumai, C. Robert, J. Astruc, J.L. Borderon and D. Flovet, "A Simulator for Intensive Care Unit Patient Data Management System". Proceedings of the Third Annual IEEE Symposium on Computer Based Medical Systems, p 454-461, 1990

[Galletta, 1989] D.F. Galletta and A.L. Lederer, "Some Cautions on the Measurement of User Information Satisfaction", Decision Sciences, 20 (3), p. 419-438, 1989.

[Gashnig, 1983] J. Gashnig, P. Klahr, H. Pople, E. Shortliffe and A. Terry, "Evaluation of Expert Systems: Issues and Case Studies in Building Expert Systems", F. Hayes-Roth, D.A. Waterman and D.B. Lenat, p. 241-280, Addison-Wesley, Reading, Massachusetts, 1983.

[Hollnagel, 1989] E. Hollnagel, "Evaluation of Expert Systems", in Topics in Expert System Design Methodologies and Tools, G. Guida and C. Tasso, p. 377-416, Elsevier (North-Holland), Amsterdam, 1989.

[Holyoak, 1989] K.J. Holyoak and P. Thagard, Analogical Mapping by Constraint Satisfaction, Cognitive Science, 13, p. 295-355, 1989.

[Ives, 1983] B. Ives, M.H. Olson and J.J. Baroudi, "The Measurement of User Information Satisfaction", Communications of the ACM, 26 (10), p. 785-793, October 1983.

[Kairouz, 1994] J. Kairouz, A. Lam, A. S. Malowany, F. Carnevale, R. D. Gottesman, "A Vital Sign Monitoring System for a Pediatric Intensive Care Unit", Seventh Annual IEEE Symposium on Computer-Based Medical System, 1994.

[Lam, 1993] A.K. Lam "Development and Evaluation of an Expert System for an ICU", Master's thesis, McGill University, Montreal, Quebec, Canada, January 1993.

[Liebowitz, 1986] J. Liebowitz, "Useful approach for evaluating expert systems", Expert Systems, 3 (2), p. 86-96, April 1986.

[Liou, 1990] Y.I. Liou, "Knowledge Acquisition: Issues, Techniques, and Methodology", Proceeding of the ACM SIGBDP Conference on Trends and Directions in Expert Systems, p. 212-236, Orlando, Florida, November 1990.

[Lucas, 1986] H.C. Lucas Jr, Information Systems Concepts for Management, third edition, McGraw-Hill, New York, 1986.

[McKerrow, 1988] P. McKerrow, Performance Measurement of Computer Systems, Addison-Wesley, Sydney, 1988.

[O'Keefe, 1989] R.M. O'Keefe, "The Evaluation of Decision-Aiding Systems: Guidelines and Methods", Information & Management, 17(11), p. 217-226, November 1989.

[O'Keefe, 1987] R.M. O'Keefe, O. Balci, and E.P. Smith, "Validating Expert System Performance", IEEE Expert, 2 (4), p. 81-90, Winter 1987.

[O'Leary, 1990] T.J. O'Leary, M. Goul, K.E. Moffitt, and A.E. Radwan, "Validating Expert Systems", IEEE Expert, 5 (3), p. 51-58, June 1990.

[Petroni, 1991] M. Petroni, N. Fumai, C. Collet, K. Roger, C. Yien, A. Malowany, F. Carnevale and R. Gottesman, " An Automatic Speech Recognition System for Bedside Data Entry in an ICU", Proceedings of the Fourth Annual IEEE Symposium on Computer Based Medical System, p358-365, 1991.

[Potter, 1990] M.C. Potter, D.S. Osherson, E.E. Smith, An Invitation To Cognitive Science: Thinking, The MIT Press, p. 3-33, 1990.

[Rai, 1989] A. Rai and A. Mendelow, "Effectiveness of information systems revisited", Proceedings of the 20th Annual Meeting of the Decision Science Institute, p. 578-580, New Orleans, Florida, November 1989.

[Rich, 1991] E. Rich and K. Knight, Artificial Intelligence, 2nd edition, McGraw-Hill, New York, 1991.

[Roger, 1992] K. Roger, "A Nursing Workload Manager for a Patient Data Management System", Master's thesis, McGill University, Montreal, Quebec, Canada, February 1992.

[Saab, 1995] E. Saab, "A Database for an Intensive Care Unit", Master's thesis, McGill University, Montreal, Quebec, Canada, October 1995.

[Sackson, 1990] M.V. Sackson, A. Varanelli Jr, and D. Baugher, "Expert Systems Validation", Proceedings of the Annual Meeting of the Decision Sciences Institute, vI, p. 367-375, San Diego, California, November 1990.

[Sharda, 1988] R. Sharda, S.H. Barr and J.C. McDonnell, "Decision Support System Effectiveness: a review and an empirical test", Management Science, 34 (2), p. 139-159, February 1988.

[Stefik, 1982] M. Stefik, J. Aikins, R.Balzer, J. Benoit, L. Birnbaum, F. Hayes-Roth and E. Sacerdoti, "The Organization of Expert Systems, A Tutorial", Artificial Intelligence, 18, p. 135-217, 1982.

[Sun, 1993] K.W. Sun, "A Nursing Workload Scheduer for a Patient Data Management System", Master's thesis, McGill University, Montreal, Quebec, Canada, March 1993.

[Turban, 1988] E. Turban, "Review of Expert Systems Technology", IEEE Transactions on Engineering Management, 35(2), p. 71-81, May 1988.

[Turing, 1950] A.M. Turing, "Computing Machinery and Intelligence", MIND - A Quarterly Review of Psychology and Philosophy, LIX(236), p. 433-460, October 1950.

[Waterman, 1986] D.A. Waterman, A Guide to Expert Systems, Addison-Wesley, Reading, Massachusetts, 1986.

[Yien, 1993] C.Yien, "Vital Sign Monitoring for a Patient Data Management System in a ICU", Master's thesis, McGill University, Montreal, Quebec, Canada, July 1993.