A Database for an Intensive Care Unit Patient Data Management System

Nicola Fumai

B. Eng., (McGill University), 1989

Department of Electrical Engineering
McGill University
Montréal

April, 1992

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

ISBN 0-315-74943-1

Abstract

Computerization has had a large impact on hospital intensive care units, allowing continuous monitoring and display of physiological patient data. Treatment of the critically ill patient, however, now requires assimilating large amounts of patient data.

Computers can help by processing the data and displaying the information in easy to understand formats. Also, knowledge-based systems can provide advice in diagnosis and treatment of patients. If these systems are to be effective, they must be integrated into the total hospital information system and the separate computer data must be jointly integrated into a new database which will become the primary medical record.

This thesis presents the design and implementation of a computerized database for an intensive care unit patient data management system being developed for the Montreal Children's Hospital. The database integrates data from the various PDMS components into one logical information store. The patient data currently managed includes physiological parameter data, patient administrative data and fluid balance data.

A simulator design is also described, which allows for thorough validation and verification of the Patient Data Management System. This simulatoı can easily be extended for use as a teaching and training tool for PDMS users.

The database and simulator were developed in C and implemented under the OS/2 operating system environment. The database is based on the OS/2 Extended Edition relational Database Manager.

Sommaire

L'informatisation des unités de soins intensifs a permis une surveillance médicale constante et l'affichage graphique des données physiologiques des patients. En contre partie, le traitement des ces patients impose maintenant l'étude d'une vaste quantité de données.

Une solution pour diminuer ce fardaud serait d'utiliser des logiciels qui accompliraient automatiquement l'acquisition et le traitement des données et offriraient des comptes rendus imprimés sur papier ou affichés sur écran selon des formats facilement compréhensibles. Des systèmes experts pourraient aussi servir à analyser les données et à suggérer un diagnostic ou un traitement. Mais afin que ces logiciels soient efficaces, il faudra les intégrer au système informatique central de l'hôpital et relier les informations diverses dans une base de données qui servira de dossier médical pour chaque patient.

Cette thèse présente la conception et la réalisation d'une base de données informatiques, développée pour le système de gestion de données médicales (SGDM) en cours de réalisation à l'unité de soins intensifs de l'Hôpital pour Enfants de Montréal. L'intégration des données des différents modules du SGDM permettra de gérer les données médicales telles que les paramètres physiologiques, les équilibres liquidiens et les données administratives des patients. Ce mémoire décrit aussi un simulateur qui a pour but d'aider à la mise au point, à la vérification et à la validation du SGDM. Ce simulateur pourra aussi être utilisé dans la formation des futurs usagés du SGDM.

Les logiciels de la base de données et du simulateur ont été développés en C dans le cadre du système d'exploitation multitâche OS/2. La base de données a été réalisée avec l'aide du gestionnaire de base de données relationnelles (Relational Database Manager) de OS/2 version"Extended Edition".

Acknowledgements

I wish to express my thanks to my supervisor, Alfred Malowany, for giving me the opportunity to experience the challenges of life in a research environment, and for allowing me to discover the exciting field of patient data management systems. His enthusiasm for the PDMS project and the inspiration, stimulus and guidance he provided through numerous discussions have been invaluable. Franco Carnevale and Dr. Ron Gottesman of the Montreal Children's Hospital also deserve special mention for their thoroughness and helpful collaboration. On the financial side, the support of NSERC, FCAR, CEFRIO and the IBM-McGill cooperative project is also gratefully acknowledged. A thank-you must also be expressed to the undergraduate students who have contributed to the PDMS project, especially Brian Romanchuk, Dave Sauntry and Mario Pagliarulo.

At the conclusion of this long yet rewarding endeavour, I remain indebted to my family and friends who have provided the support and understanding I needed to accomplish this task. Special thanks go to my "veteran" colleagues Christian Collet, Jean Drolet, Jean-François Panisset, Marco Petroni, Mathieu Pilon and Kathleen Roger. A special mention to Mathieu for guiding me through my graduate studies experience. It would also be impossible to forget my fellow graduate students, Brian Jung, Dimitrios Lambidonis, Mike Moscovitch and Frank Di Giuseppe. Equally unforgettable are the afternoon espressos at Cafe Molise and AC Milan (thank-you Marco!).

Finally, last but not least, my family must be congratulated and thanked for their support and guidance and for bearing with my absence, spiritual if not physical, for long periods. This work is dedicated to them, my parents and brother, to whom I owe everything.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

One of the most dramatic examples of medical science at work is the Intensive Care Unit (ICU) at a modern hospital. Based on the connection of patients to the latest electronic equipment, including an increasing number of computers, ICUs often mean the difference between life and death to the critically ill.

Originally developed from the post-operative recovery room, where the patient was observed during the period immediately following an operation, intensive care units provide a system whereby the continuous monitoring of acutely ill patients enables a swift response to any adverse changes in the patient's condition. In an ICU, every key factor affecting recovery is under constant scrutiny and an overall watch is kept by a computer; programmed to evaluate and compare every reading fed into it by the monitoring instruments, the computer system flashes a warning to medical personnel whenever measurements show undesirable variations. Such a constant monitoring by a computer system allows any change to be instantly dealt with.

The advent of computerized, medical equipment in the ICU, however, has led to an increase in the quantity of data requiring analysis by the medical staff. Manually managing such quantity of data can pose a heavy burden on the medical personnel, especially in an ICU, where prompt and appropriate reactions are essential. This has created the need for a means of properly accumulating, managing and assisting in the interpretation of this data in a manner which is meaningful to the health care professionals.

Unless a computerized, automated data management solution is provided, the benefits of the sophisticated medical equipment will be lost. What is required is more than just a computerized data logging system. From automated data acquisition, to electronic data archiving, all the while promptly communicating critical data, the ICU computer system must relieve the complete data management burden (data collection, data analysis, and data storage) from the medical staff, in order to be beneficial.

The costs of such computerized data management systems may be high, but can be recovered. For one, relieving the medical staff of this great flow of information allows them to tend to more important matters, improving their effectiveness. Secondly, the past two decades has witnessed a rapid development in computer technology, with a remarkable reduction in cost and size of computers and substantial improvements in speed and power, thus making the computer solution seem even more attractive. But most importantly, the computerized ICU data management system allows a concentration of patients with the most potentially lethal conditions to receive specific critical care which permits an increased number of lives to be saved.

The next sections will expand in greater detail the topics presented here. First, a historical overview of ICUs is presented, focusing on the presence of automation and technology. Section 1.2 then illustrates some of the needs of the modern ICU and how a computer-based data management system can be used to satisfy them. Subsequent sections detail some of the areas where computerization can be effectively applied (e.g. charting, bedside terminals, diagnosis) to provide assistance. Section 1.3 then presents a summary of other applications of computers in the medical field ranging from imaging to the use of computers in teaching. A snapshot of other non-medical applications where computerization has provided a solution to managing the large data flow is also provided. The final section of this chapter then gives an overview of this thesis, which presents the design of a database for an ICU patient data management system.

1.1 Evolution of the Intensive Care Unit

The evolution of specialized units in which expert medical, nursing, and technical staffs were provided with equipment for monitoring and for immediate life-saving interventions paralleled advances in invasive surgical and medical procedures. As early as the 1860's, "recesses" or "small rooms" leading from the operating theater began to appear [Nightingale, 1863]. Here patients would remain until they had recovered from the immediate effects of an operation. Such recovery rooms, adjacent to operating rooms and used mainly for postoperative care, have been described as the direct antecedents of intensive care units [Hilberman, 1975]. It wasn't until almost 100 years later, however, when a 1947 anesthesia study reported a highly significant reduction in postoperative morbidity and mortality in patients who were the beneficiaries of specialized post-anesthetic care, that post-operative recovery rooms achieved widespread acceptance [Ruth *et al.*, 1947].

From the post-operative recovery rooms developed at the Johns Hopkins Hospital in the 1920s [Kirschner, 1930], to the first shock unit (developed in Los Angeles in the late 1950's and which featured the use of invasive cardiovascular monitoring) [Weil *et al.*, 1976], to the first coronary care unit (established in Kansas City in 1962, and featuring continuous electrocardiographic monitoring) [Day, 1963], special care units have continued to evolve, both in number, degree of specialization and equipment used [Balcerak, 1990]. A study by Pollack [Pollack *et al.*, 1987] has shown that over the past few decades, development of intensive care units has followed several paths, according to traditional specialties (such as medicine and surgery), organ systems (such as cardiac, neurological and respiratory), diagnosis (such as trauma, burns) and type of patient treated (such as neonatal or pediatric).

Despite the different categories, critical care and the exceptional demands it places on physicians and nurses remains the same. Extraordinary skill is required of the medical staff, along with prompt, accurate treatment decisions. The Textbook of Critical Care [Weil *et al.*, 1989] defines the critically ill patient as requiring a complex and time-critical therapy. Physicians and nurses collect a large amount of data through frequent observations, testing, and data recorded by continuous monitoring equipment. If this data is not presented in a compact, organized form, physicians can miss important events or trends. Patients requiring therapeutic decisions to be made in minutes are often the norm, not the exception. In these circumstances, physicians need quick access to all the relevant data concerning the patient. Yet, with many of the patients, the medical record is very often not available, not up to date, or even non-existant in the case of new patients or neonates.

With an increased demand for cost-effective care and accountability by patients, utilization review committees, third party payers, and health care policy makers [Leyerle *et al.*, 1990], physicians and nurses caring for the critically ill must keep complete and accurate records [Gardner, 1986b]. In face of strong economic pressures, hospitals also have strong incentives to know the cost of procedures and to control these costs [Lyons and Gumpert, 1990].

A 1986 Statistics Canada report shows that ICU costs (currently at $1.03 billion, or 8 % of all inpatient hospital costs) and their use is growing. The significance of these numbers is exemplified by the fact that ICU cost effectiveness has received scant attention, due mainly to unavailability of data [Ellis *et al.*, 1990].

Computerization

Bloom [Bloom, 1986] reports that hospital computerization began reaching significant levels in the late 1970s. Initial computer interests involved mainframes and centralized data banks, mostly used by MIS departments. The subsequent maturity of databases led to the Hospital Information System [Scherrer *et al.*, 1990]. This permitted hospitals to consolidate common administrative data which was shared by many departments.

At the same time, innovative developments in medical instrumentation were taking place [Selby, 1990]. With computerization and electronic technology, new medical devices provided health care professionals with enormous amounts of data to process. Much of this data was lost, however, as data logging for charting and interpretation purposes was performed manually.

As the price of computers decreased, while their computational power and functionality increased, hospitals began placing computers in many departments [Abrami and Johnson, 1990]. This new wave of computerization was an effort to manage the data provided by enhanced medical instrumentation. The introduction of the microcomputer (PC) in the early 1980's began placing computers on desktops throughout the hospital. Soon software specialized for different departments, such as the laboratory, the radiology department, and pharmacy began to appear. An Albrecht and Lieske study [Albrecht and Lieske, 1985] claims that the microcomputer started a decentralization of computer resources. More and more departments began computerizing without necessarily following any plan of global sharing of computer resources [Denger *et al.*, 1988]. Sivak [Sivak *et al.*, 1989] documents that the move (of computer resources) to the desktop, rather than successfully solve the data management problem, has introduced a new one — lack of coordination and cooperation.

With these considerations in mind, one clear area of concern for the ICU is data management. Whether its use is destined for clinical decision-making, or for cost-effectiveness decisions, the data management problem is a growing concern — one that may even be hampering the medical staff from providing efficient and proper care [Nolan-Avila and Shabot, 1987]. With computer-based data management systems, the data management burden may be alleviated. The next sections will examine how a computer system can be introduced into an ICU, and the data management tasks that it can perform.

1.2 ICU Computer Data Management

Over the past fifteen years, a large number of computer-based data management systems have been developed and installed in hospitals. Many of these installations have been in the Intensive Care Unit (ICU). Gardner [Gardner *et al.*, 1989b] identifies four functions of computers in an ICU setting:

- Physiologic monitoring
- Computers that facilitate the timely and accurate communication of data among multiple hospital locations and departments
- Management of medical records
- Expert computer systems to aid in patient care decision making.

Andreoli and Musser [Andreoli and Musser, 1985] identify two types of computer applications for patient care: patient monitoring systems, and computer-assisted diagnosis systems. Kriewall and Long [Kriewall and Long, 1991] characterize computer-based medical systems as controllers, information managers or diagnostic tools. All these definitions overlap to comprehensively describe the realm of computer-based medical systems or PDMSs.

1.2.1 Computer Charting

A patient's medical record (also referred to as a chart), has constantly been the main repository for a patient's medical data. In its role as the central focus of the care process, however, the handwritten medical record suffers from certain deficiencies. As early as 1873, in her book "Notes on a Hospital" [Nightingale, 1863], Florence Nightingale wrote: "In attempting to arrive at the truth, I have applied everywhere for information, but in scarcely an instance have I been able to obtain hospital

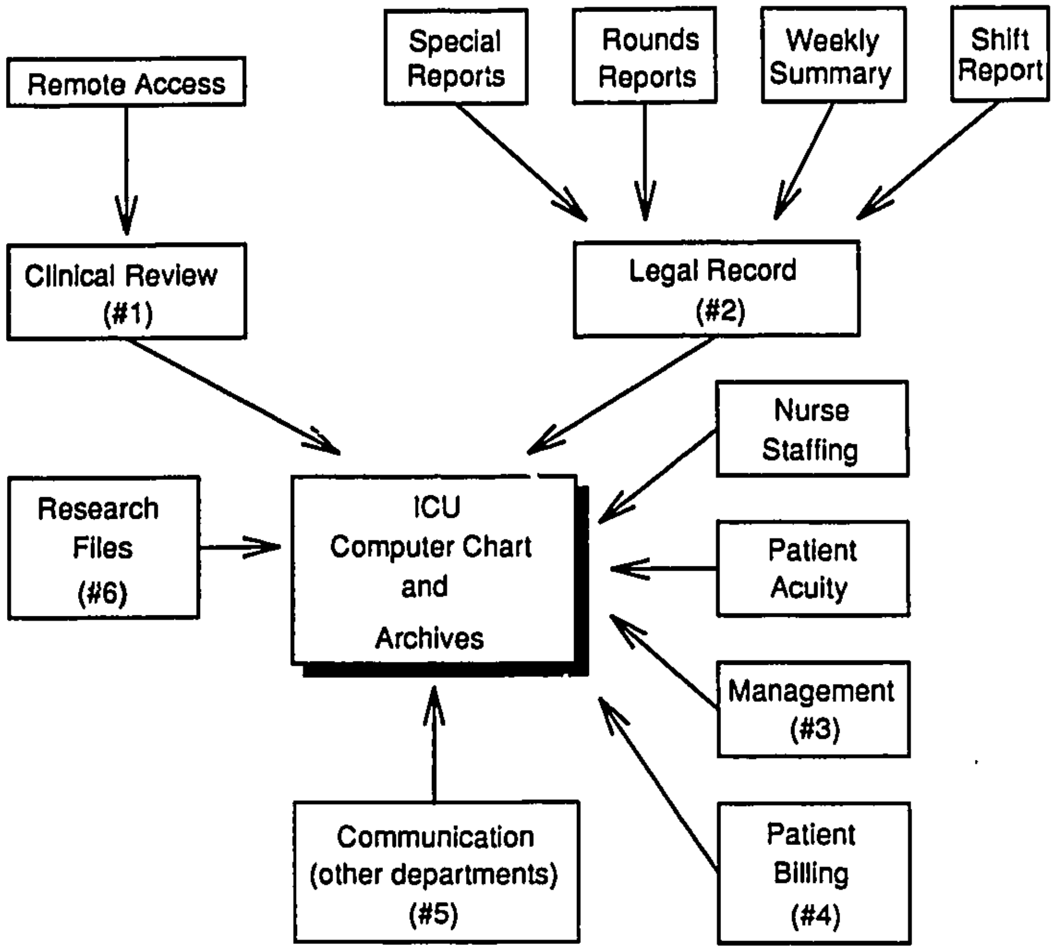


Figure 1.1: ICU Chart Complexity

records fit for any purpose of comparison. If they could be obtained, they would enable us to decide many other questions besides the one alluded to." Although much has changed in the hospital from 1873 to now, it is surprising that Florence Nightingale's comments are still applicable more than a century later.

The Textbook of Critical Care [Gardner *et al.*, 1989a] lists the major issues involved in ICU charting as data acquisition, data display and presentation, and data storage. These are illustrated in Figure 1.1, detailing the complexity involved with each issue. The patient record (chart) must record all relevant patient data and document the action taken by the medical staff. This acquired data must then be properly organized for medical and legal requirements (#1 and #2 in Figure 1.1), and for use in management (#3, Figure 1.1) and billing (#4, Figure 1.1) purposes.

Communication (#5, Figure 1.1) to other departments within the hospital is mandatory. The chart data must then be properly archived to fulfill legal requirements. The archived data is also of essential use in research (#6, Figure 1.1).

The patient's medical record is thus the principal instrument for patient care. A Whiting-O'Keefe [Whiting-O'Keefe *et al.*, 1985] study, however, shows that the conventional medical record does not offer adequate support for proper patient management. Bradshaw [Bradshaw *et al.*, 1984] indicates that this (lack of adequate support) is particularly evident in places such as the ICU, where advanced techniques of patient monitoring produce large volumes of data for evaluation and critical decisions have to be made quickly and usually at a time when data are incompletely understood. Fries [Fries, 1972] has shown that in complicated cases, the conventional record is less helpful than a structured flow chart. Several attempts have been made to bring order to the process of record keeping and improve the paper chart [Fries, 1974] [Whiting-O'Keefe *et al.*, 1980]; the traditional medical record, however, still suffers from severe limitations. A Stead [Stead *et al.*, 1983] study shows that now more than ever, because of the voluminous amounts of patient data being gathered, there is a real need to optimize and facilitate medical data review and decision making.

Let us detail the limitations of the handwritten paper record, and examine how functions in each of the three domains (data acquisition, data management, data storage) can be enhanced via computerization.

Data Acquisition

As a result of their medical problems, patients in ICUs are monitored with complex bedside monitors, and are often connected to a variety of other electronic instruments. Since the patient data is paper based, the medical staff must periodically read the data from the on-line instruments, then write this data into the chart, either plotting it as points or entered as numbers in tables which must be accumulated for meaningful interpretation. A Hammond study [Hammond *et al.*, 1991b] suggests that this procedure is tedious and error prone, because of the stressful nature of the ICU environment. Furthermore, taking a single or a few readings at a given time may not reflect accurately the behaviour over the previous time interval.

Dasta [Dasta, 1990] mentions the important role that computerized data acquisition can play in ICU environments. A computer based system can interface directly with the instruments and the data can be obtained and stored in its native, electronic form. This eliminates the data transcription errors and burden. Data accuracy will be improved since the computer system can continuously monitor and log patient data, allowing the capture of data which would otherwise be lost.

Many examples of computer-based acquisition systems can be found in the literature. The IEEE Engineering in Medicine and Biology Society Annual Conference [Nagel and Smith, 1991], the IEEE Symposium on Computer-Based Medical Systems [Bankman and Tsitlik, 1991] and other conferences have tracks dedicated to data acquisition. Some sample implementations include a system for automating alarm data acquisition in the post anesthesia care unit developed by Aukburg [Aukburg *et al.*, 1989]. Alesch [Alesch *et al.*, 1991] describes a system based on a PC/AT which contains 64 parallel data acquisition channels for acquiring data from neurosurgical monitors.

User Interfaces for Data Entry

Computer charting in the ICU must support multiple types of data collection to be effective. Tachakra [Tachakra *et al.*, 1990] shows that a large portion of the data needed to make decisions are from manual tasks such as administering a medication or auscultating breath or heart sounds. Kari [Kari *et al.*, 1990] shows that for computer charting systems to be successful, data acquisition must support manual data entered from nurses and physicians at the bedside.

User acceptance of data entry systems hinge on the effectiveness of the person-machine interface provided. In the ICU environment where computer data entry and retrieval must be done as quickly and efficiently as possible, Savona-Ventura [Savona-Ventura *et al.*, 1990] suggests that what will inevitably and eventually determine the acceptance of a system by its end users is the quality and effectiveness of its user interface. A Holtermann study [Holtermann *et al.*, 1990] also reports that when medical personnel is faced with a system which is harder to use or requires more work than the current manual methods, the staff will reject the new system. Furthermore, the computer-based system must not unwittingly force the medical staff to double chart, as this too is unacceptable [Korpman, 1990].

While remarkable improvements have been achieved in computers and in their processing speed, only modest improvements have appeared in input and output devices. The most common user interface on computers remains the keyboard [Edmonds, 1990], with pointing device variants also gaining popularity [Myers, 1991]. Murchie and Kenny [Murchie and Kenny, 1988] compare different input modalities (keyboard, light pen, voice recognition) with respect to speed, errors, and the number of corrections required to enter patient admission data. This study finds the keyboard the fastest, most accurate, easiest and most preferred method of data entry. Nevertheless, many new devices and variants have appeared on the market, in the attempt to better perform the data entry task.

Soligen and Shabot [Soligen and Shabot, 1988] present a 32 key keyboard consisting of both numeric key entry and multiple function keys which allows single-key operation of a patient data management system.

King and Smith [King and Smith, 1990] present a touch screen for system operation and entry of information, which has permitted an increase in the ability of casual users to input data effectively. Others have tried manual variants such as bar code readers [Hammond and Stead, 1988], pocket computers [Large, 1990], and even palmtops and hand held keypads [Stoeckler and Ellis, 1989]. Each of these has met with various amounts of success.

A number of new input modalities has brought new interest to the data entry problem. The most common in research circles is speech input. Martin [Martin, 1989] presents the usefulness of speech in user interfaces versus the traditional keyboard and mouse input modalities. Read et al [Read *et al.*, 1990] survey various speech analysis systems comparing capabilities and requirements. Some successful attempts at integrating speech input with data entry include:

- A speech recognition system for the entry of bone scintigram reports. Ikehira et al [Ikehira *et al.*, 1990] found that the voice entered reports were more concise and clear than the handwritten or typed equivalent.
- Fieldman and Stevens [Fieldman and Stevens, 1990] report on a study on the application of speech recognition to record dental examination data. They find that speech recognition entry typically required more time, although no difference in error rate was found between the two methods.
- Diethrich [Diethrich, 1988] describes a system in a cardiac ICU in which both speech recognition and speech generation is used to allow the nurse to enter data using voice. This allows the nurse to attend to the patient and document the treatment simultaneously.
- Petroni et al [Petroni *et al.*, 1991] describe a speech interface to facilitate the manual data entry of fluid balance data in an ICU. The speech recognition system is being investigated as a "hands-free", "eyes-free" alternative to keyboard data entry.

Whatever the input modality, each has its distinct advantages. Cohen [Cohen, 1990] reviews the latest technologies which will enhance the medical professional's ability to interface with computer systems, including bar-reading, graphical devices, and expert systems. The conclusion is that there is no one 'better'

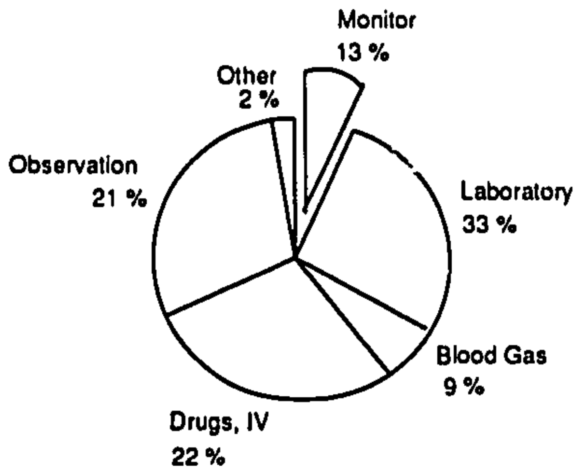


Figure 1.2: ICU Chart Use

input device, but that the optimal system will provide different input mechanisms to accommodate changing user and/or user entry requirements.

Data Presentation

A study conducted by Bradshaw [Bradshaw *et al.*, 1989] showed the importance of having a unified medical record. The study reported that when making treatment decisions doctors most frequently used laboratory data (42 %), then drug and fluid balance data (22 %) and clinical observation (21 %) and finally bedside physiological monitor readings (13 %) (Figure 1.2). These findings provide evidence that data from several sources, not just from the traditional physiological bedside monitoring devices must be communicated and integrated into a unified medical record to permit effective decision-making and treatment in the ICU.

An advantage of the handwritten chart is that it provides quick and convenient data entry, with little or no training. The computerized chart must improve the current paper charting process all the while retaining its current strong points. Bishop [Bishop, 1990] documents some of the enhancements a computerized chart can provide as being:

1. Physical Availability

   The medical record is often physically unavailable, since it can only be used by one person at one location at a time. Continuity of care is also an issue as patients are generally served by a team of physicians, nurses, and therapeutists and their data often requires transferring from one individual to another. By contrast, a computerized record, stored in electronic form, is available on demand, and can be assessed and viewed from many locations within or external to the hospital.

2. Organization

   The paper medical record is often poorly organized, available only in the order it was recorded, and many times illegible. These qualities make retrieving information from present charts a slow, error prone process [Gardner, 1986a]. In comparison, patient data stored in a computer chart may be organized to match many needs. Graphical on-screen displays can better highlight trends in patient data [James *et al.*, 1990]. Printed reports can provide quick and legible patient summaries [Dasta, 1990]. The electronic patient data may be searched or queried for specific data content [Haug *et al.*, 1990]. Hypertext links provide customized browsing features [Chaney *et al.*, 1989], while multimedia enhancements permit integration of data of various nature such as taped voice reports [Goldberg *et al.*, 1989], image data [Brown and Krishnamurthy, 1990], numerical data from monitors, and motion video of surgical procedures [Chnadrasekhar and Price, 1989].

3. Data Retrieval

   Using the paper chart for data retrieval is both cumbersome and time-consuming, since it must be done manually. This impedes the physician from finding relevant data quickly, as well as making data collection for research purposes a lengthy and expensive task. Computerized data management provides all the benefits of computer database services. Medical staff can perform complex queries on the patient data [Safran *et al.*, 1990]. Trends and anomalies on a cross section of patients can be identified with data analysis tools [Leyerle *et al.*, 1990], while networked data resources provide quick access to varied sources of data [Price and Chandrasekhar, 1989].

4. Communication

   Communicating patient data to other departments in the hospital is usually performed by sending the medical chart. This often results in numerous delays, in addition to the physical availability problems already mentioned. Computerized records, however, allow patient data to be communicated and shared via electronic means.

   Local area computer networks allow implementation of distributed data resources [Price and Chandrasekhar, 1989]. Inter-networked departments allow access to remotely stored data [Bubolz *et al.*, 1989]. Physicians may remotely access clinical and administrative information from their offices or homes [Joslyn, 1989], while wide area networks spanning countries and continents bring data and researchers together [Scherrer, 1990].

Dasta [Dasta, 1990] also mentions the important role that computers play in ICU environments, assimilating, processing, and displaying large amounts of data in easy-to-understand formats for the users.

The evaluations and comparison of computerized systems versus the manual data management systems is discussed in two articles by Tachakra et al [Tachakra *et al.*, 1990] and by Kari et al [Kari *et al.*, 1990]. In these articles, studies demonstrated that the computerized method of data acquisition and storage yielded significant savings in time, staff workload, and in accuracy. Both studies also showed that patient information was faster to access and to interpret by the physicians using the computerized systems than by using the manual paper charts.

In studies of computer-based patient monitoring and data management systems done by Hammond et al [Hammond *et al.*, 1991b] [Hammond *et al.*, 1991a], results showed that the use of these systems not only reduced non-nursing related work, but also improved the quality, quantity, and recall of the clinical information by the staff. This leads to a significant improvement in documentation flowsheets both in terms of quantity and accuracy which can in turn have medicolegal and quality assurance impacts, as well as enhancing patient care.

Data Storage

Equally important is the proper storage and archiving of patient data. Not all medical staff can view on-line data as is it is acquired. Often, important trends are only visible after a sufficient quantity of data has accumulated. An article by Schwid et al [Schwid *et al.*, 1990] discusses the uses of data acquisition for the purposes of off-line data analysis and clinical research.

Archiving issues must also be considered, since past patient data must be available for consultation, whether it be clinical, legal or research issues. Ferguson [Ferguson *et al.*, 1989] describes a microcomputer which acquires heat and fluid balance data of severely burned patients via a serial RS-232C interface to a measuring device. The computer does not monitor the patients, but the data acquired is archived in a data base and used for future predictions of heat and fluid losses, given burn area and severity.

Gibbs [Gibbs, 1989] comments on the use of computers to acquire and store anesthesia and ICU information as crucial for legal issues. The failure of meaningful, supportive, or exculpatory documentation raises serious questions about the quality of the care rendered. For this reason, Gibbs states that the use of computers for automated data acquisition and record keeping in anesthesias and ICUs is as crucial as the use of data recorders in aviation.

Savona-Ventura [Savona-Ventura *et al.*, 1990] discusses how the computerization of patients' medical records facilitates storage. The current practice of warehousing the paper-based records is impractical in both cost and efficiency. Archiving electronic data, however, is much more effortless. Recent storage developments, such as optical disks, allow the equivalent of 600 000 pages of printed information to be stored on one disc. Parkin [Parkin *et al.*, 1990] presents a description of how electronic storage will ease the warehouse archiving problems.

The use of computers in critical care management seems to offer the most obvious solution to ever increasing availability of data generated by advanced techniques of patient monitoring and a need for their immediate evaluation followed by critical decisions. In addition the apparent advantages of a computer-based system need not be limited to retrieval, presentation and analysis of complex patient data [McDonald and Tierney, 1988]: the incorporation of computer-based advisors [Padjen and Salasidis, 1988] could extend the usability of an information system in a decision making process.

1.2.2 Bedside Terminals

Patient care systems that were introduced into hospitals in the 1970s were structured around terminals at the nursing station. Most information was entered by clerks who worked at the nursing station. Other health-care providers accessed the computer through these same terminals. This arrangement was quite acceptable for the time since most computer related activities involved accessing patient administrative information from the Hospital Information System.

Abrami and Johnson [Abrami and Johnson, 1990] suggest that the idea of bedside terminals, while having been around for a number of years has only recently, with advances in networking and database technology and cheaper hardware, become economically feasible. Russler [Russler, 1991] indicates how the bedside terminal completes the computer-based data management system and acts as a platform for the integration of the various medical data sources. The bedside terminal may also actively participate in the distributed data network.

Halford et al [Halford *et al.*, 1989] identify several advantages which bedside terminals can provide:

1. Reduced duplication of data

   A Hammond et al study [Hammond *et al.*, 1991a] found that with centralized terminals, 82 percent of nurses first wrote down data, then later entered it into the computer. Bedside terminals, however, would permit physicians and nurses to enter or review patient data immediately.

2. Improved completeness and accuracy of data

   Being able to record information quickly, without having to leave the bedside will also simplify the entry of data.

3. Reduced time spent waiting for central terminals

   For both data entry and review, bedside terminals allow immediate access to the patient data. Rathe [Rathe *et al.*, 1989] states that bedside terminals will provide "point of use" access to patient and clinical data.

4. More sophisticated displays

   By exploiting the power of computerization the patient data can be graphed, plotted, tabled or listed in a variety of ways. The "as is" restriction of paper-based charts is loosened.

5. Improved retrieval capabilities

   The bedside terminal provides access to all the electronic patient data, from parameter data to lab results. This provides on-site exploration of physiological trends and fluid balance data. The bedside terminal permits both nurses and physicians to "research" patient problems as they encounter them on rounds. The physician can quickly find the result or a piece of information needed to resolve an acute problem.

6. Patient care planning and documentation

   Bedside terminals can play an active role in the patient care process, assisting nursing staff to ensure that the right activity is performed on the right patient and is properly documented. Bedside terminals also provide the perfect setting for computer-based Nursing Care Plan and workload manager systems [Roger *et al.*, 1991].

There are some concerns though: additional cost, effect on patient (noise and light may be distracting or annoying), ergonomics of terminals, patient confidentiality and data security, lack of space and infection control all must be taken into account [Hendrickson *et al.*, 1991].

1.2.3 Computer-Aided Decision Support

Using computers as a decision aid is not new. Since the first detailed suggestions by Ledley [Ledley and Lusted, 1959] and Lusted [Lusted, 1968] computer-based applications have appeared which provide diagnostic aid to physicians. These early applications also met with certain success. It isn't until recently, however, with more powerful technology, new algorithmic approaches, and increased availability of computerized ICU patient data that the field of computerized diagnosis and decision support has expanded. Recent years have seen computerized medical decision making gain wider acceptance, so much so that the discussion of artificial or machine intelligence is common place in medicine today [Blum and Semmel, 1991] [Miller and Giuse, 1991].

The main concerns with the physician's current diagnostic process lie with the data acquisition and data processing functions. Dombal [de Dombal, 1986] has shown that some doctors do not elicit information at all effectively from patients, asking large numbers of quite irrelevant questions, often at the expense of questions which would be relevant. As for processing, Perry [Perry, 1990] reports that there are doctors do not assimilate at all well the information they receive from patients, and ignore obvious clues for diagnosis and management. Due partly to the large patient data volume and the inability of the physician to manage it all, Durbridge [Durbridge *et al.*, 1976] indicates how some doctors obtain large quatities of relatively useless biochemical data of which they utilize less than 5 percent.

The computerized system can aid in several aspects, ranging from controlled monitoring to decision-making. The monitoring systems may either screen the medical staff from the raw, abundant physiological data or may be used in closed-loop patient control. Collet et al [Collet *et al.*, 1990] describe a system which analyses cardio-vascular data from physiological monitors to generate early warning alarms, while Albisser [Albisser, 1989] describes attempts to use microprocessors to manage diabetes mellitus through the use of closed-loop control.

The decision-making systems serve to put the large databases of electronic patient data to use by analyzing and integrating data to generate a medical diagnosis. Several methods have been used successfully, including:

- Trend detection systems which study trends in stored patient data to report diagnostic predictions [Duhamel *et al.*, 1989]
- Rule-based, artificial intelligence systems, such as that used by Varde [Varde *et al.*, 1991] for diagnosing thyroid dysfunction
- Neural network models such as that used by Ozdamar [Ozdamar *et al.*, 1989] for EEG spike detection

For the moment there is yet no clear cut widespread data to suggest which of the above methods has an advantage since all seem to work reasonably well. The underlying principle in the above systems is the efficient use of the patient data. Bradshaw [Bradshaw *et al.*, 1984] points out that if current performance levels are inadequate, with the increasing complexity of modern medicine, the difficulty in assimilating information (and consequent poor performance) will become more pronounced. Unless the patient data can be properly put to use, what good is there in acquiring it!

There are, nonetheless, still some issues left to be resolved before a full computerized decision system can be implemented on a large scale. There are criticisms over the algorithms being used [Avent and Charlton, 1990] and the procedures, if any, being used for verification and validation [O'Neil and Glowinski, 1990] [Fieschi, 1990]. The main concerns, however, involve ethical and legal issues, and practical issues. At the heart of the practical issues is the concern over the structure of the data itself [Lee and Lee, 1991]. There is no agreement about definitions of symptoms or signs, or what symptoms are to be collected — or even what disease categories there are and how they are defined [Humphreys and Lindberg, 1989]. Consequently, there is no coordination and sharing of data banks, causing the task of computer-aided decision support to be magnified several times.

1.3 Computer-Based Data Management Systems

1.3.1 Patient Data Management Systems

Both commercial and in-house hospital-developed PDMSs have been successfully implemented. Factors influencing choice of commercial versus hospital-developed include: ability to customize, functionality, cost, user interface and multi-vendor equipment support [Paganelli, 1989].

The commercial systems include:

- The Hewlett Packard PDMS/CareNet system [Hew, 1988]
- The EMTEK system [EMT, 1988]
- The MedTake system [Pesce, 1988]
- The Health Care Solution by the LGS Group, based on the TDS 7000 series software by TDS HealthCare Corporation [Gro, 1991]
- The Ulticare Modular information system developed by the Health Data Sciences (HDS) Corporation [Hea, 1991]

- The computerized patient chart which uses pen-based notepads being developed by Purkinje Inc. [Sys, 1991]
- The SIDOCI (Système Informatisé de Données Cliniques Intégrées) system [SID, 1991]

The in-house systems are developed primarily in universities, medical schools and hospitals. Some sample implementations include systems such as the Health Evaluation through Logical Processing (HELP) computer system at Latter Day Saints Hospital (Utah) [Bradshaw *et al.*, 1989] or the workstation-based system for use in critical care environments described by Prakash [Prakash *et al.*, 1991].

An integrated information management system is also being developed at the Hôpital Notre-Dame in Montreal. This system has modules for integrating patient admission, radiology reports, diatetic data and pharmacology reports into the computerized patient chart. The chart can also be viewed at any workstation. A module for outpatient appointment management has also been developed [Hop, 1991].

The Hôtel-Dieu de Quebec Hospital has developed a sophisticated care and workload planning system. The PSI (Plan de Services Integrés) can also integrate with other medical information systems to permit integration of reports and data from other departments. The PSI software is distributed through the Hypocrat company [Les, 1991].

1.3.2 Other Medical Systems

The computerized approach to data management has been applied to other fields of the medical profession. Osler et al [Osler *et al.*, 1990] describe a system to manage a data bank of injury descriptions. The system integrates both data entry and database searching for specific injuries, classes of injuries or combinations of injuries. The injury description system is intended for widespread application in academic trauma centers where an accurate and accessible trauma database is important.

Sinclair and Ross [Sinclair and Ross, 1990] report on a recombinant DNA laboratory manager, based on a PC using DBASE III+. The system coordinates such functions as keeping track of patient samples, blot membranes, polymerase chain reaction products and test results. A data entry mechanism is also available via a menu driven interface. The PC-based system saves time and reduces data handling errors compared to the manual method, as well as provide the ability to rapidly look up data and produce customized worksheets or reports correlating all available clinical and laboratory information.

Goldberg et al [Goldberg *et al.*, 1989] describe a computerized workstation and integrated multimedia communication link between a radiology department and an emergency department. The system is being implemented at the University of Ottawa Medical Communications research Center to improve communications between radiologists and attending physicians. The multimedia system consists of several workstations linked by the hospitals' local area network. The work stations are 80386 based PCs with a 500 Mbyte image disc and a 1000 line monitor. The multimedia data comprises digitized images (2430x2000x10), voice data (via a PBX and Meridian Mail digitized voice messaging system) and graphics.

1.4 Overview of Thesis

This thesis presents a database design and implementation for the Patient Data Management System (PDMS) being developed jointly between McGill University and the Montreal Children's Hospital. The PDMS intended use is the pediatric ICU ward of the Montreal Children's Hospital. The database design presented here is an integral part of the PDMS system, as data storage, archiving, and retrieval availability issues are key to proper data management. The PDMS database, relying partly on the OS/2 Extended Edition Relational Database Services, organizes the patient data in a relational format, and allows full retrieval capability via Structured Query Language (SQL).

The next chapter presents an overview of the PDMS system and the hardware and software environment in which this system will function. The various PDMS modules are presented, along with a discussion of some of the design and implementation issues involved. Emphasis is placed on module functionality and overall program structure rather than on low-level software details.

Chapter 3 focuses on the database and data storage issues, outlining the design philosophy and the implementation issues. The functional architecture of the database is presented in detail, and a discussion of the interaction between PDMS, user and database are presented. Topics covered include the data organization, internal data storage structure, and the overall software architecture.

Chapter 4 discusses the PDMS/CareNet simulator, developed mainly for debugging and test purposes. The focus is on outlining the interactions between the PDMS and the simulator. The capabilities of the simulator are discussed as well as areas of deficiency. A presentation of how simulation capabilities can be added is also given, since future testing of the PDMS system will be an important concern.

Chapter 5 reproduces some sample sessions with the PDMS. Data storage and retrieval from the PDMS database is benchmarked, allowing us to evaluate the performance of the design and suggest improvements. Other extensions are presented which will enhance the PDMS system. Lastly, the conclusion summarizes the material presented in this thesis.

Chapter 2 The PDMS System

This chapter formally describes the Patient Data Management System (PDMS) being developed at McGill University for the Pediatric Intensive Care Unit (PICU) of the Montreal Children's Hospital. The PDMS is a computer-based information management system for handling patient data in the hospital's pediatric intensive care unit. The PDMS must accommodate the needs of patients ranging from newborns to adolescents, handling data measured electronically with physiological monitoring equipment, data measured at the bedside, and data made available from other departments such as laboratory results. The function of the PDMS is to monitor and record all patient data, to facilitate the review and interpretation of the data by presenting CRT displays of trends, plots and charts, and to assist in the hardcopy documentation requirements by generating useful forms and required reports.

The PDMS system is an attempt to improve the efficiency and accuracy of patient management by addressing both information and knowledge management in the ICU. Efficiency can be improved by cutting time spent in information recording, retrieval and evaluation; the thoroughness and accuracy of the data is also improved, as well as allocation of resources [Halford *et al.*, 1989]. Computer-based advisors and expert systems can contribute substantially in the decision making process in the ICU [Gardner, 1986b], while scoring systems can help predict the outcome of a population of patients in the ICU. With a precise prediction of patient outcome, management of resources can also be improved as this will provide individual evidence of the benefits/requirements of intensive care [Knaus and et al, 1985].

The PDMS system is an ongoing project, with new enhancements and functionality continuously being added. This chapter begins with an overview of the

PDMS, and its operating environment. The following sections then expand the hardware and software environment to greater detail. In these sections, the entire PDMS system is presented, with an aim towards overall functionality, and not implementation specifics.

2.1 PDMS Overview

The design and development of the PDMS system began almost ten years ago, when a need for some computerized help in the ICU was realised. The early 1980's had seen a dramatic increase in the number of medical monitoring devices available on the market. The increase in the amount of data available to the medical staff had resulted in an increased amount of time required to document and analyze this additional information. It was soon realized that in most cases a computer could do the task of data collection and documentation faster and more accurately than a human [Carnevale, 1987].

Many commercial systems have been available for a number of years. These off-the-shelf products offer the advantage of being ready to install in their entirety with little involvement from the medical staff, which contrasts to the long development cycle of a home-grown system with its many staff consultations and try-outs. There are, however, drawbacks to these commercial systems. Most commercial systems available are not customized for a specific installation. Customization may be difficult and costly. As well, the hospital may be paying for functions they do not need and will not use. This increases the cost and complicates the user interface.

Most commercial systems used in ICUs have met with very limited success mainly because of a poor interface design. The principle criticism by nurses and physicians of these commercial packages is their lack of user friendliness [Hendrickson *et al.*, 1991]. Also, the lack of involvement of the users in the initial planning causes serious limitations, as software packages could not be later corrected or modified. All of these factors contribute to resistance to the technological innovations available for ICUs, making commercial PDMSs come short of fulfilling their initial objective of relieving the workload of nurses and doctors.

For a patient data management system to be successful, the system must satisfy the functional requirements identified at its inception, it must be able to integrate new functions or interact with external computing facilities, and its user interface must be perceived as responsive, unobtrusive and intuitive. User attitudes to computerization and to the patient data management system must also be considered. An examination of a successful data management system, Adamo, in the Netherlands [Hasman *et al.*, 1988] attributed the success of the system to the fact that it contains only those functions that are really necessary.

The commercial systems' failure to address all the data requirements properly resulted in the decision to have the PDMS developed in-house.

2.2 The Hardware Environment

The hardware configuration of the PDMS is presented in Figure 2.1. It is based on the HP CareNet system which provides a local area network for linking multiple HP 78534A physiological monitor/terminals to a HP 78581A Network Systems Communication Controller.

To gather the information continuously generated by each bedside monitor, all bedside monitors are linked together using the CareNet network. The bedside monitors are connected in a star configuration with the System Configuration Controller (SCC) at its center, collecting the transmitted information. To access the SCC, the presence of the CarePort interface unit between the host computer and the SCC is required. As far as the host computer is concerned, only the CarePort unit exists; the host computer has no knowledge of the functioning or even existence of the SCC. The HP 78580A CarePort unit is a programmable interface which

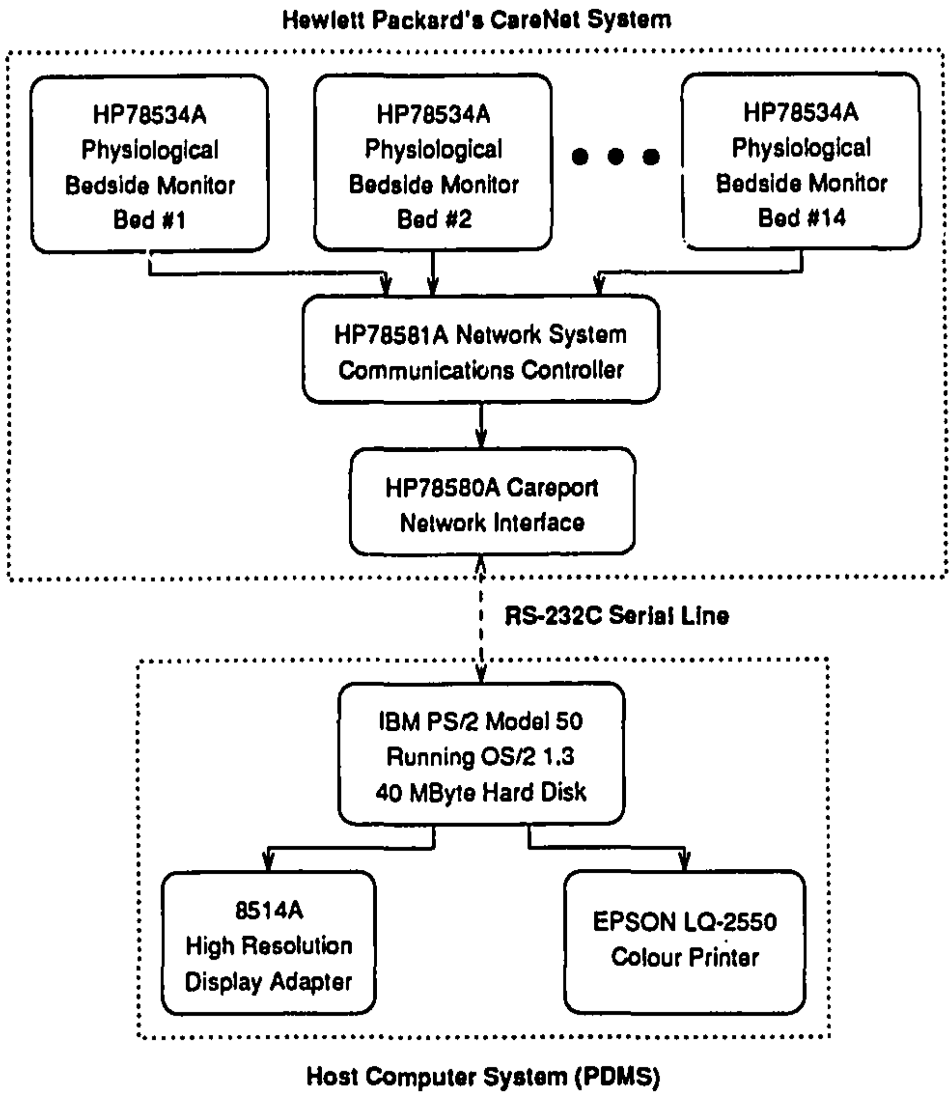


Figure 2.1: PDMS Hardware Configuration

translates the commands of the host computer into the proper format for the SCC and communicates the results or the selected bedside information back to the host computer. The CarePort unit and the host computer communicate with each other via a standard serial RS-232-C interface.

The PICU setup includes fourteen bedside monitors which have been in operation for over two years. In addition to acquiring physiological patient data, these monitors also provide stand-alone data management capabilities including automatic smoothing of measured parameters and real-time display of selected data. The patient data acquisition capability can be further enhanced with optional plug-in measurement modules and a keypad for manual data entry.

The host computer presently consists of an IBM PS/2 model 50 with an 80286 processor, 5 Mbytes of memory, a 30 Mbyte hard disk, an 8514 colour display, and running version 1.3 of the IBM OS/2 operating system. To this will soon be added an IBM PS/2 model 80 computer with an 80386 processor, 8 Mbytes of RAM memory, a 150 Mbyte hard disk, and an 8514 high resolution colour display. The two systems will be networked using a Token-Ring network, and the IBM OS/2 LAN server/requester network software. The model 80 PS/2 will function as the PDMS server, dedicated to acquiring and storing the patient data. The model 50 system will be used mostly as a front-end system for the interactive PDMS interface such as the trends module and the expert system. The initial fault-tolerance constraint of the PDMS design (which designated both systems to be run in parallel) can now be loosened. With the new database implementation, database recovery is no longer a problem and as discussed in chapter 3, the fault-tolerance burden can be shared by the two systems by exploiting the network connection.

2.3 The Software Environment

2.3.1 Operating System Considerations

The early PDMS system was prototyped for use on an IBM Personal Computer (PC), based on an Intel 80286 microprocessor and operating under control of the MS-DOS operating system. The initial design consisted mainly of a routine which acquired the physiological patient data and displayed this using a simple character-oriented display on a monochrome screen.

After the first prototype, feedback from the hospital staff indicated that the major drawback was the text-based trend displays, which generated cluttered graphs. A new PDMS design was proposed [Panisset *et al.*, 1989] which revised the user interface almost entirely. This new design adopted the graphical user interface, based on the new, emerging Window-Icon-Mouse-Pointer standard. The new design also made use of the latest enhanced graphic color displays.

While the MS-DOS operating system may have provided sufficient services for the early PDMS system, it offered no support for graphical interfaces, windowing systems, modular system design, and extended memory management. The limitations of MS-DOS became a burden, making the software environment too stringent to support the growing complexity of the PDMS program.

Although more expandable environments were available with other hardware configurations (e.g. UNIX workstations) it was decided to remain with the PC mainly because the PC environment was the least expensive in terms of hardware, yet still allowing easy changes and configuration expansion. The 1988 release of OS/2 by IBM and Microsoft brought all the benefits of the larger hardware environments to the PC platform. OS/2 provides programs with benefits such as large real memory, virtual memory, memory isolation, I/O protection, resource management, and multitasking services.

Due to the numerous benefits of a multitasking operating system, it was decided to modify the design of the PDMS to implement it under OS/2. Clearly, instead of having to design the PDMS as a single program to perform all the necessary tasks, it is now possible to write separate programs to implement the required functions such as network data acquisition, graphical trend display, and fluid intake/output management. Each of these modules runs separately, and the interfaces among them are defined in terms of such operating system primitives as pipes, semaphores, and shared memory segments, thus minimizing the possibility of harmful interaction among the modules. Furthermore, it then becomes easy to add more modules as the design progresses.

2.3.2 User Interface

The overall organization of the various PDMS modules into a coherent system presenting a consistent user interface was dictated at least partly by the constraints of OS/2 version 1.0, the version of OS/2 currently available at design time.

OS/2 introduces the concept of a screen group, or session, which is a collection of processes that share a single virtual screen, keyboard and mouse. The process — OS/2's highest level of multitasking — refers to an instance of the execution of a program. The screen group is a logical grouping or collection of processes, making them share the same display, keyboard, and mouse. A screen group can be either in the foreground (foreground session), or in the background (background session). When in the foreground, all video output from any of the programs belonging to this group instantly appears on the screen, and programs within the group have immediate access to input from the keyboard or mouse. A background screen group's processes continue to run even though the screen group is not visible. Video output generated by these background processes is routed to a Logical Video Buffer (LVB), which is later redrawn on the screen when the screen group is brought to the foreground. Background processes are thus unaware of whether their output is currently visible, but rather continue to send video data to a virtual screen (the LVB). Likewise, background processes use a virtual keyboard and mouse for user input. Through this method, OS/2 allows groups of concurrently running applications to share the display without conflicts.

Implementing each module as a separate screen group allows each module to run independently and concurrently. These modules are then grouped together in a tree structure under a common "main menu". This gives the PDMS a hierarchical menu structure which facilitates its comprehension; users become accustomed to having to pass through the main menu (the System Manager menu) to access any module.

The goal of the current design of the user interface is to thus correct previous criticisms, by accommodating the personnel needs at the Montreal Children's Hospital. It is important that the format of the modules be kept flexible to allow easy modification to adapt to the future needs of the hospital. The user-interface must also be friendly and intuitive. The following goals were established for the user-interface design:

1. Easy modification and extensions. By exploiting the modularity permitted by OS/2's multitasking, modifying various components of the PDMS can be done quickly and efficiently, without extensive modification of the core PDMS routines. This is further enhanced by OS/2's support of shared Dynamic Link Libraries (DLLs).

2. Friendly and intuitive design. Most of the user input and interaction is done through the use of menus and windowed prompts. The users are not required to learn any sophisticated command syntax or mnemonic keys. For advanced users, who will use the system extensively, shortcut keys are provided.

   Where possible, the menus and user prompts were made similar to the paper based forms already in use. The staff is presented with an interactive screen they are already familiar with, which allows for less training time, decreased anxiety, and fewer input errors.

   An on-line help facility is also being designed to provide assistance and training for the users. The HELP command is to be available at any point in the program and will return the user to the same point in the executing program when help was requested. This "context-sensitive" help is important in giving users confidence and control over the system.

3. Error Recovery. In a stressful environment such as the ICU, using a program with "immediate action" keys can lead to command errors which must then be recovered. Therefore, all command keys must be followed by pressing the ENTER key for confirmation or ESC to correct it.

4. Consistent "look and feel". To make the general "look and feel" of the user interface consistent across the different modules, a single menuing procedure was developed, which can run both in text modes and graphics modes. This procedure allows the display of any number of menu items in a column or a row-column format. The cursor keys are used to navigate a color highlight which indicates the current selections. The procedure also accepts a list of "shortcut" keys, which can move the pointer directly to the desired selection without having to use the arrow keys.

5. Color and Graphics. The use of color and graphics significantly enhance the display of trends. Color allows overlapping trends to be distinguished more easily, while a graphical display of patient data helps reduce screen clutter and enhances interpretability. Tullis [Tullis, 1981] has shown that effective use of color can convey more information and reduce the monotony of tasks. Variable color settings also allow users to match the displayed colors to the workplace environment.

2.4 PDMS Modules

Figure 2.2 shows a block diagram of the PDMS software. Here, a tree structure is implemented to give the user access to the various modules of the system via a main menu. As the PDMS was originally designed for use on one, central workstation, organizing the various modules around a common main menu facilitates its comprehension. The arrows in Figure 2.2 indicate communication between modules. As outlined above, the availability of a multitasking operating system has allowed the design of the PDMS as a series of interacting modules. The modules which have been implemented as of now are:

1. The System Manager which starts up the other modules and acts as a main menu to allow users to go from one module to the next.

2. The Data Link Controller (DLC) which interfaces to the network of bedside monitors through the Careport network interface and an RS-232C link. This module is concerned with the acquisition of the real-time patient data and alarms from the network, the storage and management of these data in the PDMS database, and transmitting commands from other modules to the network (such as when adding a new patient to the network).

3. The Patient Registration Module which takes care of admitting and discharging patients and managing "administrative" patient information such as name, hospital number, etc. This information is made available to the other modules of the system.

4. The Trend Display Module which allows the health care professionals to review the data acquired from the bedside monitors as graphical trends.

5. The Fluid Balance Module which uses a spreadsheet format to interactively manage the recording of all substances which are taken in (ingesta) and put out (excreta) by a given patient.

2.4.1 The System Manager Module

To run the PDMS, the System Manager module is first started, and it takes care of starting up the other modules as "children" screen groups. Since each module is a separate screen group, only one can be active in the foreground at any one time. The Manager presents the main menu which allows users to select which module is to be used. When a choice is made, the specified module is brought to the foreground, where the user can interact with it. When the user wants to return to the main menu, the appropriate command is selected inside that particular module which then signals the Manager, who in turn will dispatch the module to the background and bring the main menu back to the foreground. The System Manager module thus acts as a central command dispatcher, switching between the

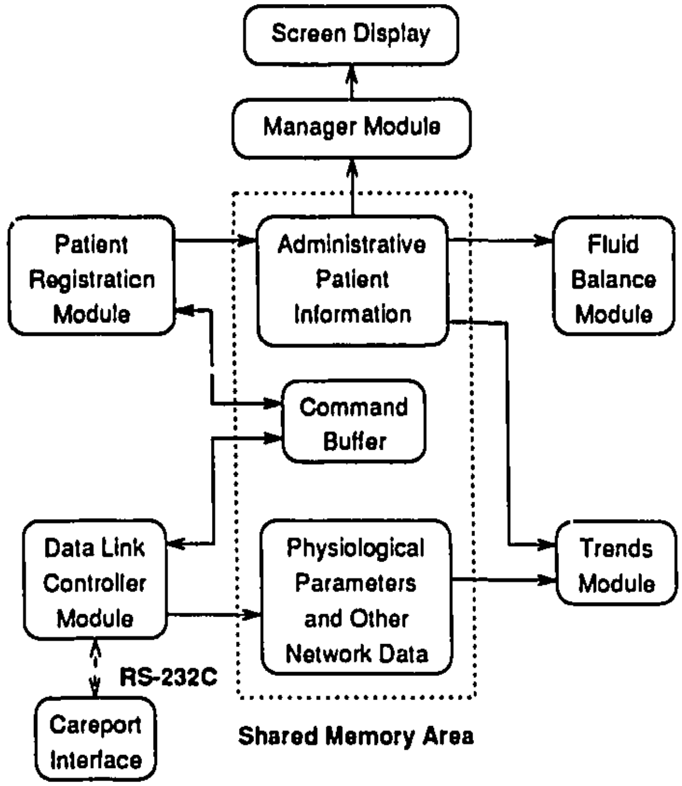


Figure 2.2: PDMS Software Modules

various modules at the user's request. This mechanism allows the user to preserve a sense of order and not feel lost in any menu hierarchy. From the Manager main menu, the user may select either of the Registration, Vital Trends, or Fluid Balance modules. The main screens of these modules offer further choices, including that of returning back to the main menu.

2.4.2 The Data Link Controller Module

The Data Link Controller (DLC) module is in charge of interfacing the PDMS with the CareNet bedside monitor network through the CarePort interface, managing the gathered information for easy access by other modules, and storing the data in the PDMS database for future consultation and archive purposes.

One of the principal functions of DLC is to obtain the monitor parameter values and to place them into circular queues after some averaging. These values can later be accessed by the Trends module for graphic trend display. Information about the network status of each bed must also be shared with the Trends and Fluid Balance modules. The parameters and other network data are therefore placed in shared memory (Figure 2.2). To avoid read-write conflicts, semaphore handshaking is implemented for parameter queues and network data access. Commands are passed to DLC through the use of a command buffer (Figure 2.2), also placed in shared memory. Semaphores are again used for access control and event signalling.

Most of the activity of DLC deals with communicating with the Careport interface. This requires the DLC module to perform multiple tasks. It must be able to receive the automatic data and alarm messages, periodically average and store the parameter values in memory or on disk, and at the same time be able to proceed with command message exchanges, some of which can modify the definition of logical sources and their data messages. This multi task implementation was solved by utilizing the pipe and multiple thread features of the OS/2 operating system. The DLC module is structured around five threads: *main*, *read port*, *alarm handler*, *accum params* and *save params*, which perform the various operations. The implementation of these threads is detailed in [Panisset *et al.*, 1989]. This multi-thread DLC implementation facilitates its programming by decoupling the various DLC functions.

2.4.3 The Patient Registration Module

This module manages the admission, activation, suspension and discharge of ICU patients from the PDMS representation of the ICU ward. It is through this module that the patients' demographic data and administrative data is entered. This module also gives the user control over network activities such as connection, activation and suspension of a bed from the network.

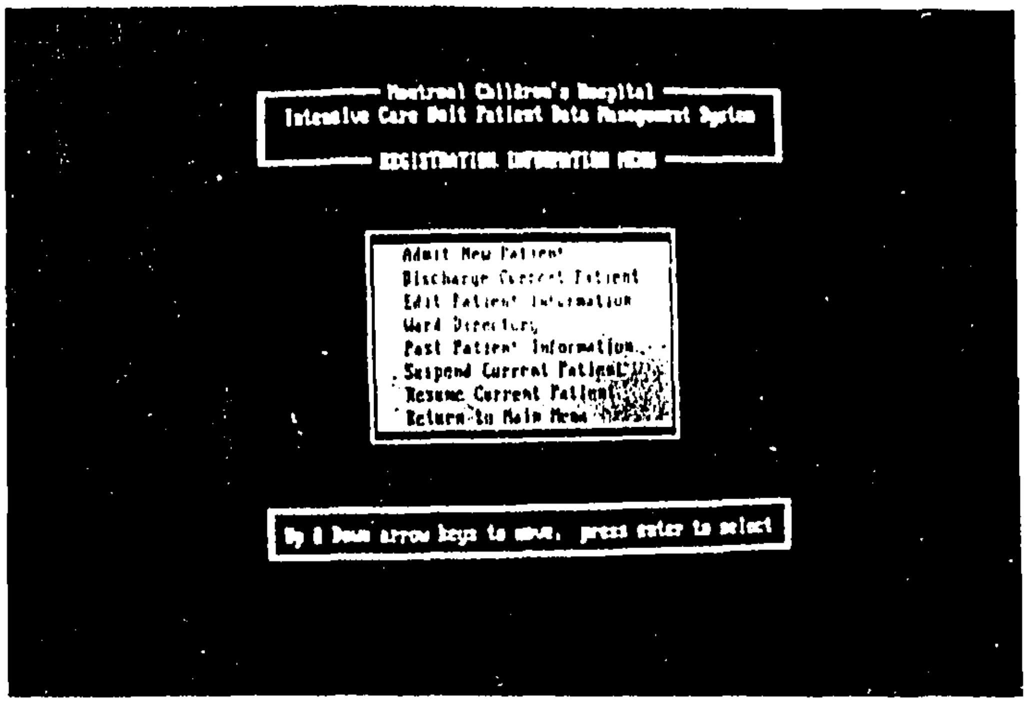


Figure 2.3: Registration Module Menu

When the Patient Registration module is selected from the System Manager main menu, the user is presented with the main Registration menu shown in Figure 2.3. The commands selectable from this menu are described below.

Admit Patient

The admission menu, Figure 2.4, is the sole vehicle via which a patient is registered into the PDMS system. Here, various text-entry fields allow the patient's administrative data to be entered. This data includes the patient's name (first and last), sex, address, telephone number, treating doctor, hospital number, ICU bed number, and date and time of admission. The ICU bed number is a concept which allows patients to be tracked on a bed basis rather than patient name or hospital ID, since in many cases, this information is not known at admission time. The bed number is also used as an indicator of available beds remaining in the ICU.

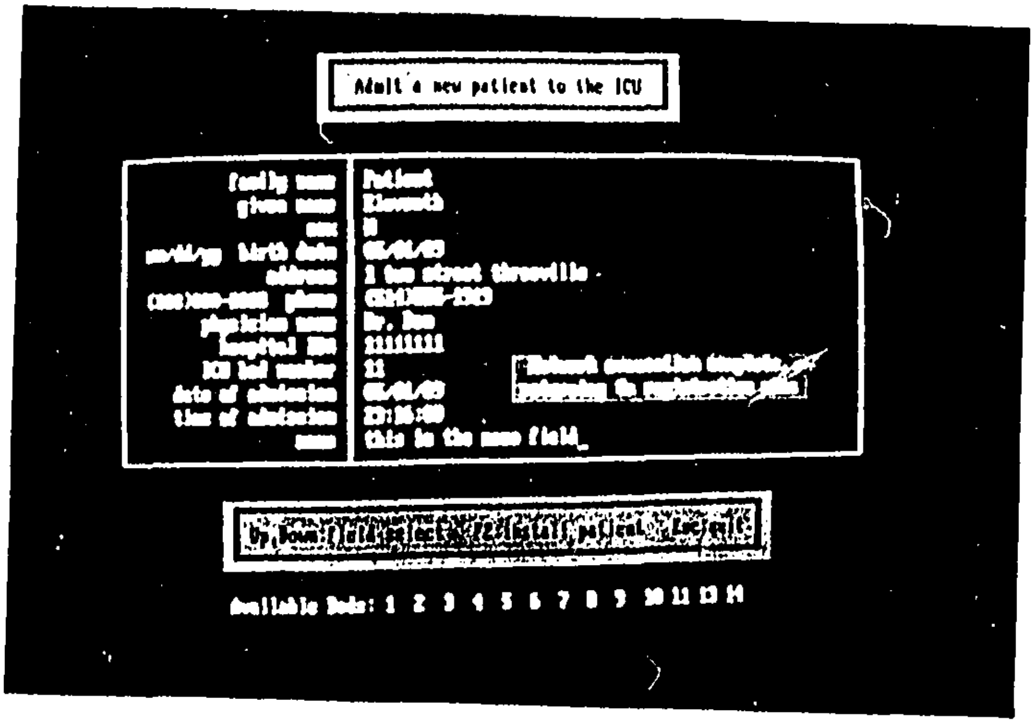


Figure 2.4: Admit Patient Menu

Much of the text-entry fields of the admission menu may be left blank initially, allowing emergencies to be dealt with immediately. The only requirements are that the hospital ID must be entered. If the admit date and time are left blank, it will automatically be set to the current date and time. A blank bed number field will let the PDMS system assign the patient to the next available free bed, in increasing numerical order. The other fields can later be completed via the edit menu.

When the field entries are completed, the patient information is saved and the patient registered to the PDMS by the 'F2' key. A command to modify the logical source definition is passed to the DLC module, via the command buffer. DLC will initiate the required transaction with CarePort and begin acquiring physiological parameter data. The DLC module will also create a shared memory structure in which all the patient information is stored and can be referenced by all the PDMS modules. The same data will also be written to the PDMS database. This is described in greater detail in chapter 3.

Edit Patient

This menu allows the patient administrative information (entered via the registration module's Admit Patient menu) to be edited. The menu is identical to that for Admit Patient, with text-entry fields which can be modified. The user can thus edit erroneously entered information or information which was left blank.

Should the user edit the hospital ID field, the patient data stored to date will be left intact for legal purposes, and a cross reference will be made to the new patient ID. The old data will be marked with a special code to prevent conflicts with old or new patient data. Other editing constraints, mainly for legal purposes are that the admittance and discharge dates and times are marked as having been edited, and the previous values are still stored.

Editing is terminated again with the 'F2' key, with the updated information written to the PDMS database. No network modification is necessary, unless the ICU bed number was modified. This may occur if the patient was moved from one bed to another. If the ICU bed number is changed, the logical source modification is again performed through the DLC module, with the same procedure as for admitting a patient.

View

This allows a quick summary report, listing patients, hospital ID and associated bed numbers of patients currently registered in the PDMS system.

Retrieve

By specifying only 'key' information, such as the hospital ID, the retrieve menu will query the database and return the matching administrative information if found. Coupled with an input device such as a bar code reader, this could allow for quick and efficient registration of patients.

Suspend/Resume

This specifies to the PDMS system to suspend or resume network data acquisition for a specific patient. This will be mainly used when the patient is temporarily transferred to another department (e.g. lab tests, XRays). Suspending the network data acquisition will prevent null data from being stored and varying the patient trends.

Discharge

This menu selection brings up an entry field where the user specifies the discharge date and time. If no date or time is specified the current date and time is used. Once discharged, the PDMS ceases acquiring network data for the patient. The ICU bed number is freed for use by another patient, and any on-line patient data is flushed to the database.

2.4.4 The Trend Display Module

The interactive display of graphical trends constitutes one of the most important aspects of the PDMS. The visual correlation of selected parameter trends is a major feature and much effort has been invested in the design of the display screen structure (Figure 2.5). A consistent use of function keys, menu highlighting and cursor control was selected to make the system easy to learn and use. It was decided to maintain a basic resemblance to the current manual charts for familiarity and user acceptance. The displays are available on a "bed basis" just as is the case for the present patient folders.

Clearly, most of the data needed in the ICU is much more relevant in graphical

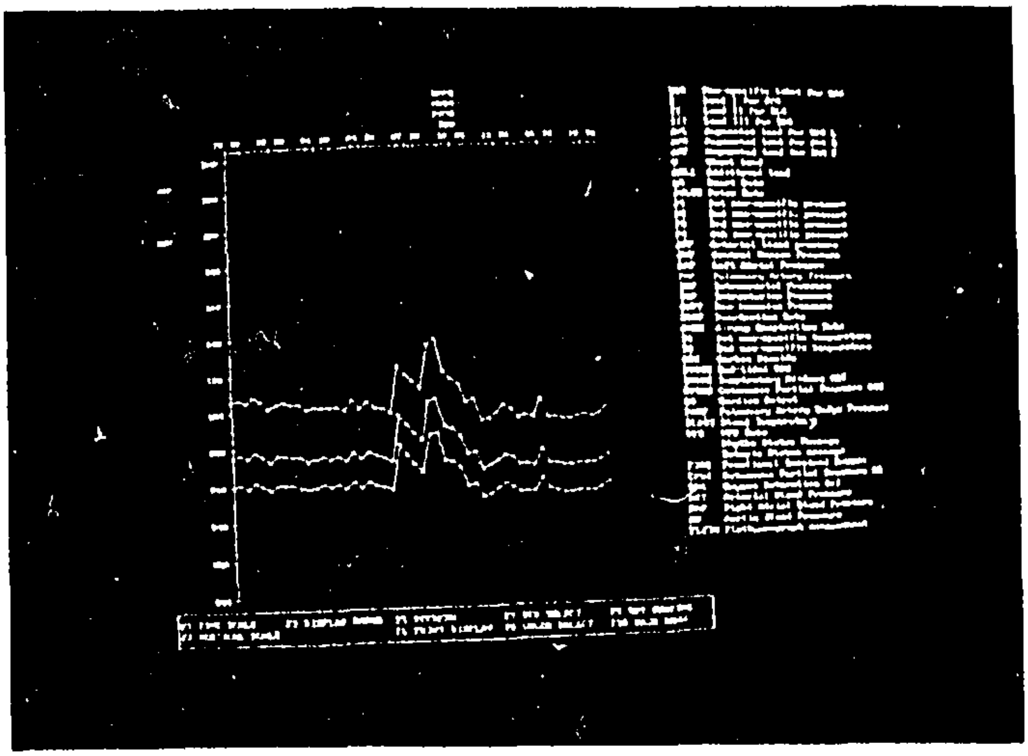


Figure 2.5: Vital Signs Trend Display

than in numerical form. For the case of vital sign data coming in from the network, the user can interactively select what data sets to display at the same time on the screen, and thus be able to observe the relations between these. Parameter data can be displayed in real-time for a remote location, or can be retrieved from storage. Once displayed in graphics format, it is easy to see whether a trend is developing or is only temporary.

Since there is only a limited quantity of data which can be represented on the screen, the screen appears as a sliding window which can be used to view any part of the trend data. There are three modes of operation: in the second data mode, the screen can display the equivalent of 50 consecutive seconds of data points, in the minute data mode the screen can display the equivalent of 50 consecutive minutes of data points, whereas in the half-hour data mode it can display 24 hours of half-hour data points (also 50). By allowing the user to specify the display range for data, the Trend Display module can display patient data in memory, and/or retrieve and display patient data stored in the database. Each possible physiological parameter is assigned a distinct color and symbol pattern combination, which allows the simultaneous display of several trends. Some trends result from triple-valued parameters (i. e. systolic, diastolic and mean values); such parameters are displayed as three traces using the same color and symbol, since by definition the three will almost never cross and will always have the same order.

2.4.5 The Fluid Balance Module

Fluid Balance is defined as the measurement of all substances (usually liquid) going in (INGESTA), and all substances coming out of the body (EXCRETA), to determine the overall balance of fluids. This has to be monitored because of the effect the balance has on blood pressure, dehydration, pooling, drowning or thrombosis.

The Fluid Balance module enables the entry, calculation and correction of the volumes of all the fluid intake and output of a given patient. The spreadsheet format of the form, shown in Figure 2.6, emulates as closely as possible the actual paper forms used in the intensive care unit by the nurses.

Keyboard data entry for the Fluid Balance module data presents a bottleneck, since there is currently only one PDMS console and fourteen beds. The ideal solution would be to have infusion pumps linked electronically to the PDMS, much in the same manner as the bedside monitors are linked through CarePort. Presently, the ICU has a variety of infusion pumps, each of which provides a different interface to the pump, making this option unwieldy.

In order to effectively integrate the manual entry of data into the PDMS and facilitate human-computer interaction, a speech interface is being investigated [Petroni *et al.*, 1991]. The voice recognition system would allow the medical staff to enter bedside data using speech.

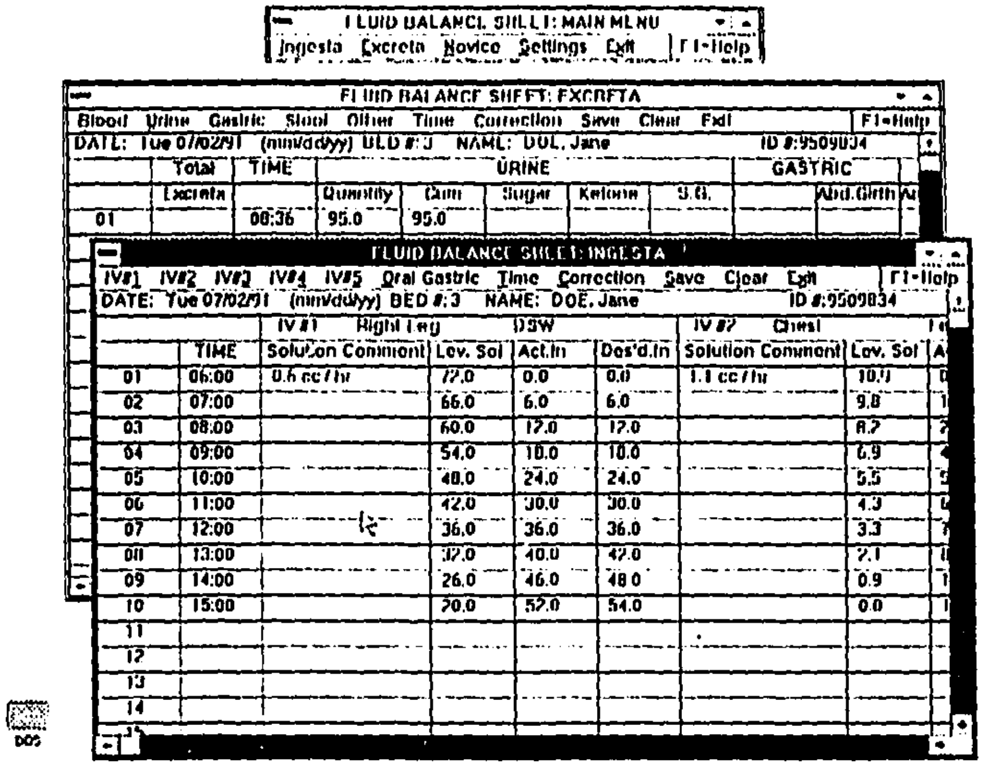


Figure 2.6: Fluid Balance Module Display

As with the Registration Module, if any fields of the Fluid Balance data sheet are edited, the previous value is still stored for legal purposes, and the field is marked with a special code to indicate that it has been edited.

Chapter 3 The PDMS Database

This chapter details the design and implementation of the PDMS database subsystem. The first section gives an overview of the OS/2 Extended Edition Database Manager. The data entities and relationships important to the PDMS are then described. Following this, the database architecture and its implementation are detailed.

3.1 The OS/2 Database Manager

Operating System/2 (OS/2) is a new operating system developed by IBM and Microsoft for personal and desktop computers based on the Intel 80286 and 80386 microprocessors. OS/2 uses the protected mode of these microprocessors which allows a large address space, virtual memory, and multitasking capabilities. In addition, OS/2 also offers operating system features such as dynamic linking, a device independent program interface, and a graphics windows-based interface. As an extension to the basic product, IBM offers the Extended Edition of OS/2, which includes the Database Manager and Communications Manager packages. These provide database management services and local area network communication services, respectively.

The Database Manager is a relational database management system (DBMS) which is fully integrated into the OS/2 environment. As a DBMS, the OS/2 Database Manager provides control of stored data, appropriate interfaces to that data, and the tools necessary for building applications that utilize the data. Unlike other DBMSs, however, the Database Manager was specifically designed for the OS/2 environment. This implies that the Database Manager can achieve a high level of functionality by using such OS/2 features as multitasking and concurrency.

The Database Manager consists of two parts: Database Services and Query Manager. Database Services provides the relational "engine" of the system, handling the tasks of data definition, storage, manipulation, and control. Database Services also provides the Application Programming Interface (API) through which programs can interact with the Database Manager. The Query Manager is a higher-level interface to the features supported by Database Services. It embodies a graphical front-end interface which acts as a buffer between the user and the lower-level functionality of Database Services. Query Manager also supplies the Customized Interface — a fourth generation language or applications generator for creating customized database applications.

The OS/2 Database Manager is a relational DBMS. This implies that the data is represented using relations. In addition to relations, Database Manager understands other database objects such as views, indexes, queries and forms. (For a detailed description of the relational database model, refer to Date [Date, 1990]).

Operations on the relations are performed through the Structured Query Language (SQL). SQL has been adopted by the ANSI institute as the standard language for relational database systems. A significant feature of SQL is that the same language is available at two different interfaces, namely an interactive interface and an application programming interface. As an interactive query language, SQL statements can be executed from the Query Manager's prompted interface. As a database programming language, SQL statements can be embedded in the high-level language source code of an application. For a complete description of the Database Manager SQL statements and functions, refer to the Database Manager programmer's guide [IBM, 1990]).

Operating under the OS/2 environment, the Database Manager supports multi-task and multi-thread concurrent access to database data and conforms to the ACID (Atomicity, Consistency, Isolation and Durability) test standards set by the

Transaction Processing Council. Consistency levels of *repeatable read, cursor stability* and *uncommitted read* are supported through the use of buffers, two-phase commit and rollback operations, and various locking levels.

The Database Manager ensures integrity and recovery of the data by providing utilities for making and restoring backups of a database. Additionally, the Database Manager keeps a log of all database transactions. Should a database become corrupted, the backup can be restored and the log can be used to bring the data up-to-date.

Security of database data is enforced in two ways. First, each database can be password protected. The password must then be supplied by any user or application program attempting to connect to the database. Second, each user is assigned an authorization or login ID. Access priviledges to the database and its objects can then be controlled for each "user" via GRANT and REVOKE operations.

In association with the Communications Manager, Database Manager provides the implementation of distributed databases across a local area network. The Communications Manager component, which supports the APPC (Advanced Program-to-Program Communications) protocol, supplies the underlying communications software for the remote services feature. With remote data services, applications can then access databases regardless of where on a network these databases reside.

3.2 PDMS data types

The PDMS is designed to store and retrieve a large amount of varied types of data. This includes both data from the network of physiological monitors and data entered manually by medical staff. Currently this latter category only includes patient administrative data, but extensions to manage other manual data can easily be added.

Network Data

Incoming data from the physiological monitor network is of four types:

1. Parameter Data are monitored data elements which can be represented numerically. This is data which originates from the bedside monitors, such as heart rate, respiration rate, or blood pressure. Parameter data, identified through a Medical Function Code (MFC), is updated every 1024 ms.

   There are two messages related to parameter data: Parameter Value (PV) and Parameter Support Data (PSD). PV messages contain the binary values for all the parameters from active selected data sources coming from a bedside monitor, whereas PSD messages contain the ASCII text displayed on the monitor for each active source. Parameter values can come single valued or triple valued. A triple valued parameter is defined by three numbers rather than one. Triple valued parameters are associated with MFC's greater or equal to 128.

2. Wave Data are digitized, sampled representations of the real-time waveforms given by the bedside monitors. This data, such as ECG and pressures, are also identified by an MFC and can be available every multiples of 32 ms. Wave data is always 125 Hz except for ECG which can also be 500 Hz. As for parameter data, values and support data are separate. The Wave Values (WV) messages contain digitized wave data samples as a function of time. The Wave Support Data (WSD) message contains wave calibration and scale information, as well as the ASCII wave label.

3. Alert Data are data strings which contain messages describing some alarm or inoperative state of a monitor. Alert data is identified by the couplet of MFC and Qualifier Code and is updated every 1024 ms. The Alert Status (AS) message contains the current status of alerts for a bedside monitor. The Alarm and Inop text (AT and IT) messages contain the text being displayed on the

monitor for the current alarm or inop. There may be multiple simultaneous alarms or inops for a bedside monitor. Because alerts remain in action often for more than a second, a change count is associated with each alarm. It indicates by remaining identical or changing its value if the alarm is being repeated or occurred for the first time.

4. Status Data corresponds to the Instrument Status (IS) message broadcasted on CareNet by each bedside monitor every 1024 ms. The status data includes the internal states of the instruments as well as a list of the waves and parameters which they are currently transmitting onto CareNet.

The parameter and wave data is marked with a validity code. A wave or parameter value is valid when the source has detected no problem with the data. Invalid data possess some detectable problem and is therefore unreliable.

Non-Network data

There is much data which does not come from the physiological monitor network. This data is available either through other channels, or is entered manually by the ICU medical staff. This includes:

- Demographic Administrative data. This includes data such as a patient's name, address, hospital ID number, telephone, birth date, social insurance number, and medical insurance (MEDICARE) number. This is data which is required for administrative purposes, and for proper identification of admitted patients. This data can be entered into the PDMS system via the Registration module Admit menu. For the PDMS, all that is required is the patients' hospital ID number. Most of the other data is probably already available via the main hospital information system (HIS). Although duplication of this data is wasteful, the communication links between PDMS and HIS have not been finalized, forcing the PDMS system to store the data.

- Fluid Balance Data This is data measuring the intake/output of fluids for a patient. This data is entered through the Fluid Balance module menu. The data is mostly numeric in nature (fluid measurements), but also contains text (fluid name or type), and date/time (date and time of measurement or administration) data.

- Medical Diagnostic Codes. To track the diagnosis of patients, various diagnostic codes are used by the medical staff. This is mostly textual data entered manually by the medical staff.

- Patient Observations. This is free text data entered by the medical staff, when it is desired to record certain patient observations or comments on the patient condition.

3.3 Database Architecture

The PDMS database will be required to serve most of the needs of the PICU. It will be accessed by both the PDMS modules and other application programs (either batch or interactive) used in the ICU. The PDMS database must be able to meet ever-changing requirements of existing or new applications. It must capture both the static (i.e. information content) and dynamic (i.e. processing operations) properties needed to support the required transactions and queries. To fulfill this role properly the following requirements must be considered during the design process:

- Availability. Data should be totally free from the constraints of any specific implementation scheme, while being available on demand to all requests that need it. Manipulation (queries, updates) of the data must follow standard, documented procedures, and must be easy to perform.

- Security. Data should be stored securely, and available only to those people or programs authorized to use it.
- Archiving. Data storage will follow consistent methods, while leaving the data in a maintainable form. Long-term archiving must balance storage efficiency and retrieval accessibility.
- Cost. The proposed implementation must consider the hardware and software costs involved. These costs must be kept in line with the intended environment.

The PDMS data architecture is organized in a multi-level structure, shown in Figure 3.1. The hierarchy contains a memory resident section (patient table and circular queues), a relational database manager with associated disk based database, and a long-term data archive section. This multi-level structure allows the PDMS data storage structure to accommodate efficient data sharing, the real-time nature of some of the data, and the voluminous data storage task, all the while remaining flexible and easy to tailor.

3.3.1 Memory Resident Data Structure

The memory resident part of the PDMS database stores data in RAM only for those patients currently in the ICU. As network data is acquired for a patient by the DLC module, the data is first stored in memory resident structures. Later, either on a periodic basis or when a patient is discharged from the ward, the data is flushed from memory and stored in the relational database on disk.

The memory resident data structure is created and initialized by the DLC module at start-up of the PDMS. All the structures are created in an OS/2 shared memory segment to allow all PDMS modules access to this data. This is necessary as various modules of the PDMS need to frequently reference and/or update the

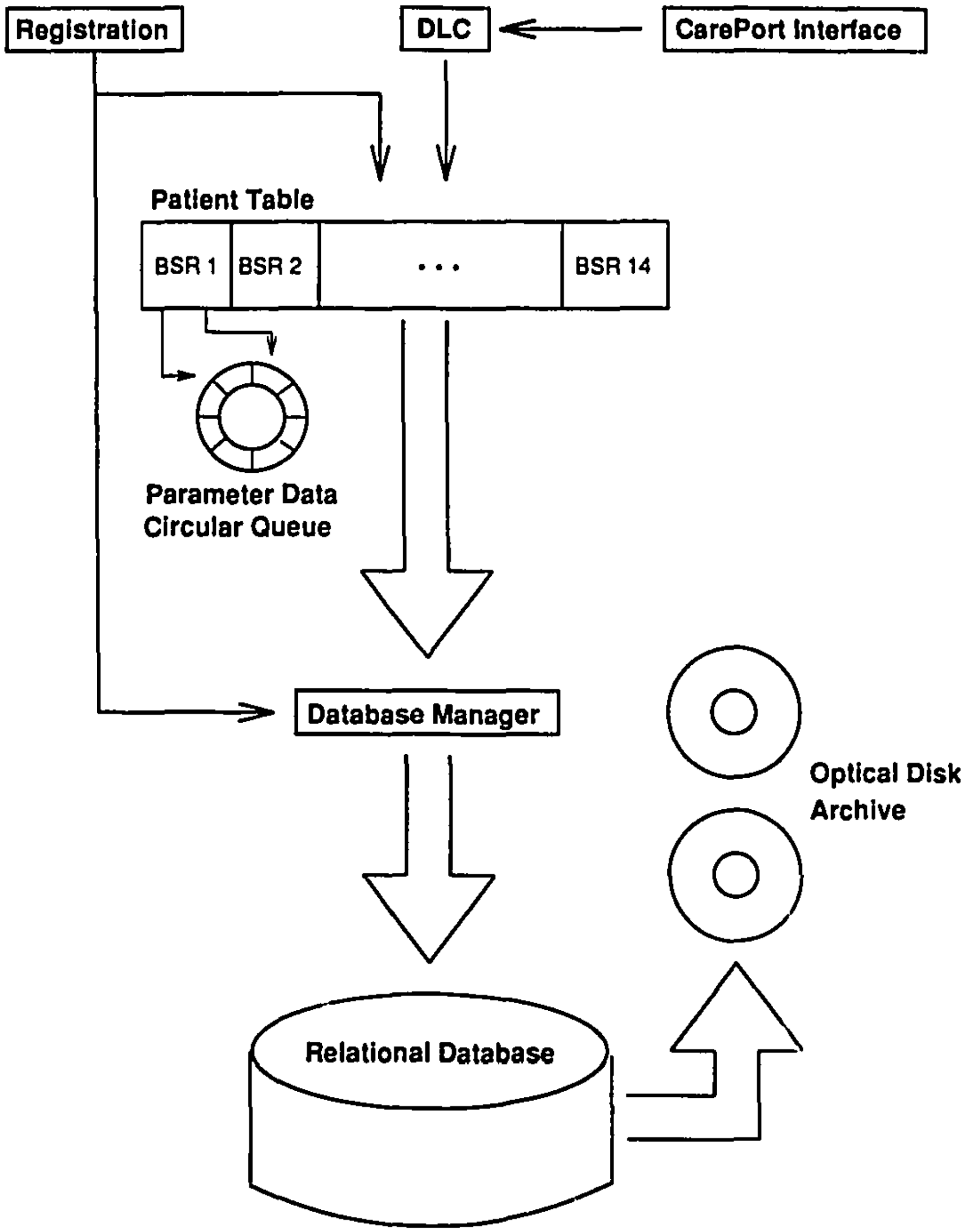


Figure 3.1: PDMS Database Structure

data fields. Two memory resident data structures exist: the Patient Table, which includes a data record for each bed, and the Parameter Data Queues, which are circular queues for holding the patient parameter data acquired from the bedside monitors.

The Patient Table

The Patient Table (Figure 3.2) is a memory resident array of Bed Status Records (BSR), indexed by bed number. Each BSR therefore corresponds to a bed in the ICU. The different fields of the BSR contain information about the patient residing in the bed. The Bed Status field indicates whether the bed is 'free', 'in-use', or 'suspended' The other data fields indicate patient name, hospital ID, attending physician and parameter data queue pointers.

The BSR fields are initialized every time a new patient is admitted into the PDMS through the Patient Registration Module. Upon registration, the patient's name, hospital ID, and bed number are entered along with other textual information. The Registration module then writes this data to the appropriate fields of the bed's Bed Status Record. The DLC module then initializes the parameter data queues and sets the bed-status field of the BSR to *in-use.* The network connection with the CarePort interface is then initiated by the DLC module.

Maintaining this patient information in memory allows the PDMS modules fast access to information used to maintain on-line displays. Furthermore, the Patient Table coordinates access to the parameter data queues. The parameter data pointer fields in the BSR are used to access the first and last parameter data points for the data queues. This provides the DLC module with a quick, direct access mechanism for parameter data saving.

This memory resident data is constructed, maintained, and read by various modules of the PDMS according to a well defined set of protocols and sequences to

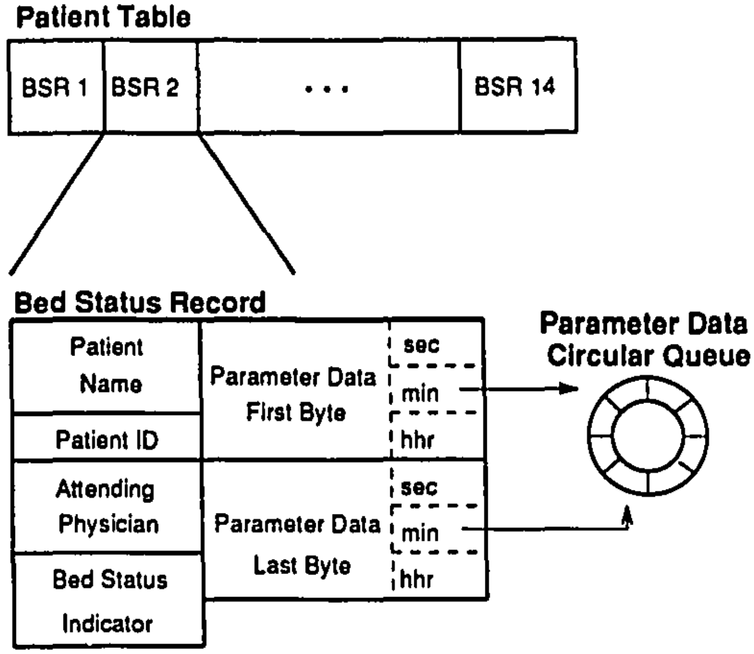


Figure 3.2: Patient Table with Bed Status Record

ensure data integrity. Each module operates on parts of the data which is relevant to its function. Interprocess communication and synchronization is accomplished by employing OS/2 services such as semaphores, pipes, shared memory segments and timers.

Data Queues

As network parameter data is acquired in real-time by the DLC module, it is stored in memory resident, circular queues. There are three such queues: a second data queue, a minute data queue and a half-hour data queue. For each active patient, parameter data is received from CarePort approximately every two seconds, stored in the second data queue, and added to 2-second data accumulators. After sixty seconds, the accumulated 2-second data points are averaged over the minute and the averaged value is placed in the minute queue and added to the minute data accumulators. The accumulated minute data points are then averaged out after thirty minutes and the value is place into the half-hour queue.

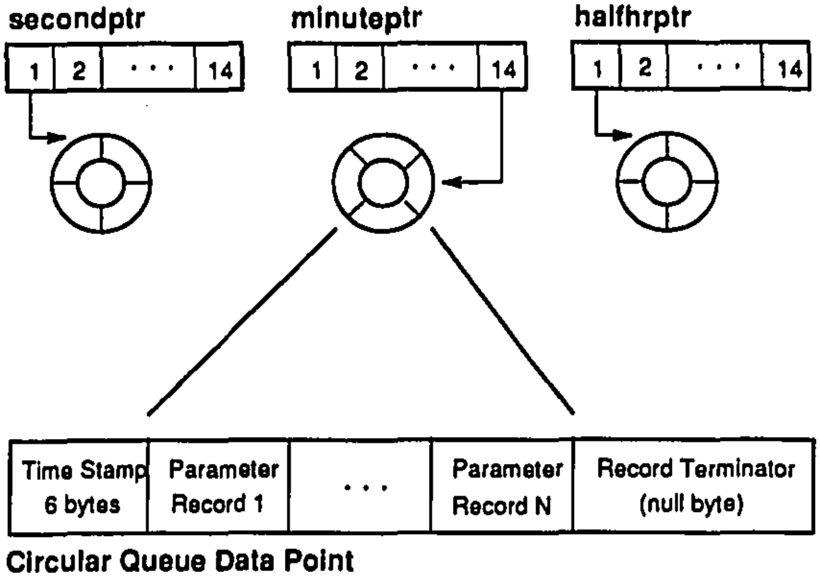


Figure 3.3: Organization of Parameter Data Queues

Parameter Indicator (type & length) 1 byte	data byte 1	data byte 2	. . .	data byte 8

Figure 3.4: Variable Length Parameter Value Record

The parameter data circular queues are organized as a three dimensional array of bytes, indexed by bed number, data point number, and data point byte number, respectively. This organization is shown in Figure 3.3. The 'secondptr', 'minuteptr', and 'halfhrptr' pointers reference arrays whose elements are indexed by bed number. Each of these bed number elements in turn references a circular data queue, whose individual data points are arrays of bytes. The individual bytes of the data point array are then accessed on a byte basis.

One queue data point is actually an array of bytes occupying a maximum of 39 bytes. A queue data point refers to an averaged sampling of multiple parameter values for one bed corresponding to a certain time interval. The first six bytes of a queue data point constitute a time stamp while the remaining 33 bytes contain the variable number of variable length parameter value records shown in Figure 3.4.

A one byte indicator at the beginning of each parameter value record indicates which parameter is stored in the record and its length. The data bytes follow. Parameter data are either single valued (2 bytes), or triple valued (6 bytes). An example of a triple valued parameter is blood pressure, where there exists values for systolic, diastolic, and mean. The last parameter value record is followed by a zero byte indicating end of the queue data point.

The current limit of 39 bytes for a queue data point implies that one queue data point can hold up to eleven single valued parameters, or four triple and one single valued parameter, or any combination in between. This limit has been arrived by studying present parameter charting at the ICU, and has been found to be adequate for present needs.

Storing the network parameter data in memory resident queues ensures that the DLC module can effectively manage the real-time data acquisition task. Thus, rather than requiring a disk access approximately every two seconds, the data is transferred to disk in blocks, at periodic intervals. The archive interval is currently set at five minutes. Every five minutes, the *save params* thread of the DLC module will write the minute and second data points to the database, along with one half-hour point every sixth five minute interval (i.e. every half-hour). The second data points are stored in a circular manner, with the new five minute block replacing the previously archived second points. This temporary storage allows the medical staff to review a patient's most recent second data. If the data points are deemed important they can be moved to permanent storage.

Memory Usage

The Patient Table contains 14 Bed Status Records, each of which may require up to 207 bytes of storage. This puts storage requirements for the Patient Table at 2898 bytes (2.83 Kbytes).

Accumulators for the 2-second and minute data are identical and require a total of 4.10 Kbytes of memory. For the circular queues, the second and minute queues hold parameter data for the preceeding two minutes and two hours respectively (i.e. 120 data points per queue, with 39 bytes per data point), for up to fourteen patients. Memory required for these buffers is 65520 bytes (63.98 Kbytes) each. The half-hour queue holds data for the same number of patients, for the preceeding two days (i.e. 96 half-hour data points per patient). 52416 bytes (51.19 Kbytes) are needed for this buffer.

The current total RAM storage requirements for the memory data structures is thus a minimal 186 Kbytes. The memory resident data structures only support administrative data and vital sign parameter data. This will have to be extended to include other data from the bedside monitor network. As support or alarms, INOPS and wave data is added, additional RAM storage may be required. Malowany [Malowany *et al.*, 1989] describes one possible organization for alarm data handling.

3.3.2 Relational Database Organization

Central to the multi-level storage structure is the physical data storage component, represented by the relational database in Figure 3.1. This is where all the PDMS data will be stored, to serve as a central repository for both the memory resident structures, the PDMS module data requests, and asynchronous user queries for patient data.

Relational Database Manager

The PDMS database is organized as a relational database, using the IBM OS/2 Extended Edition Version 1.3 relational database manager services. Previous implementations of the PDMS database have used a hierarchical structure [Malowany *et al.*, 1989] and an ISAM file/B+ trees structure. An evaluation of these systems determined that a new implementation, based on a relational model would provide several advantages [Fumai *et al.*, 1990]. Lee [Lee and Lee, 1991] has also evaluated several different data models for a medical patient data database. This study has concluded that the relational approach offers a number of advantages:

1. Data Independence.
   Operations on the data do not require using a specific, pointer sequencing, access method, inherently linked to the data file implementation. Changing the data file implementation does not require recoding of the procedures which operate on the data. In relational database systems in fact, the low-level physical implementation is hidden from the user and developer.

2. Data Concurrency.
   Relational data systems have matured in a multi-user, multitasking environment. Under OS/2, the relational database manager can make full use of multitasking offering multiple accesses and shared data facilities. With many of the other PC–based database implementations there is no support for concurrent data access.

3. Enhanced Query Capability.
   The OS/2 Database Manager supports the ANSI standard Structured Query Language (SQL). SQL uses english like syntax, enhancing the accessibility to the stored PDMS data.

4. Network Distributed Access.
   By being tightly coupled to the OS/2 system and the Communications Manager sub-system, the database will be fully networkable, allowing a distributed PDMS data structure. And, with relational database systems widely available on other hardware environments, the PDMS can easily be configured on a heterogeneous, database network.

Patient Address	Admittance Date	Parameter Name
Patient Birthdate	Admittance Time	Parameter Value
Patient Hospital ID Number	Discharge Date	Parametor Timestamp
Patient Name	Discharge Time	Parameter MFC Code
Patient Sex	Attending Physician's Name	
Patient Telephone Number	ICU Bed Number	

Figure 3.5: Important PDMS Entities

Database Relations

Date [Date, 1990] identifies two important elements of the database design process as being the conceptual and external schema definitions. The conceptual schema will describe the data and relationships of importance, while the external schema will describe how these will be modeled and implemented on any given DBMS.

The important data items for the PDMS database are listed in Figure 3.5. The relationships between these items is then shown in Figure 3 6 in entity-relationship (E/R) diagram form. This E/R diagram represents the conceptual schema of the PDMS database.

Turning to the external schema part of our database design, it is important to recall at this point that the database will be relational, thus the external representation will describe relations, attributes of these relations and so on. The relational database can be implemented by defining several relations which would group together common data. Figures 3.7 through 3.9 show the relations for the PDMS database. Here, the primary key attributes are those columns marked as 'data required'.

The ADMIN_DATA relation (Figure 3.7) will contain the patient's administrative data. Most of this data is textual in nature, and is therefore stored as characters or character strings. As the hospital ID will be unique for each patient, it is used as the primary key. The birthdate is stored as a character string for compatibility with existing PDMS modules.

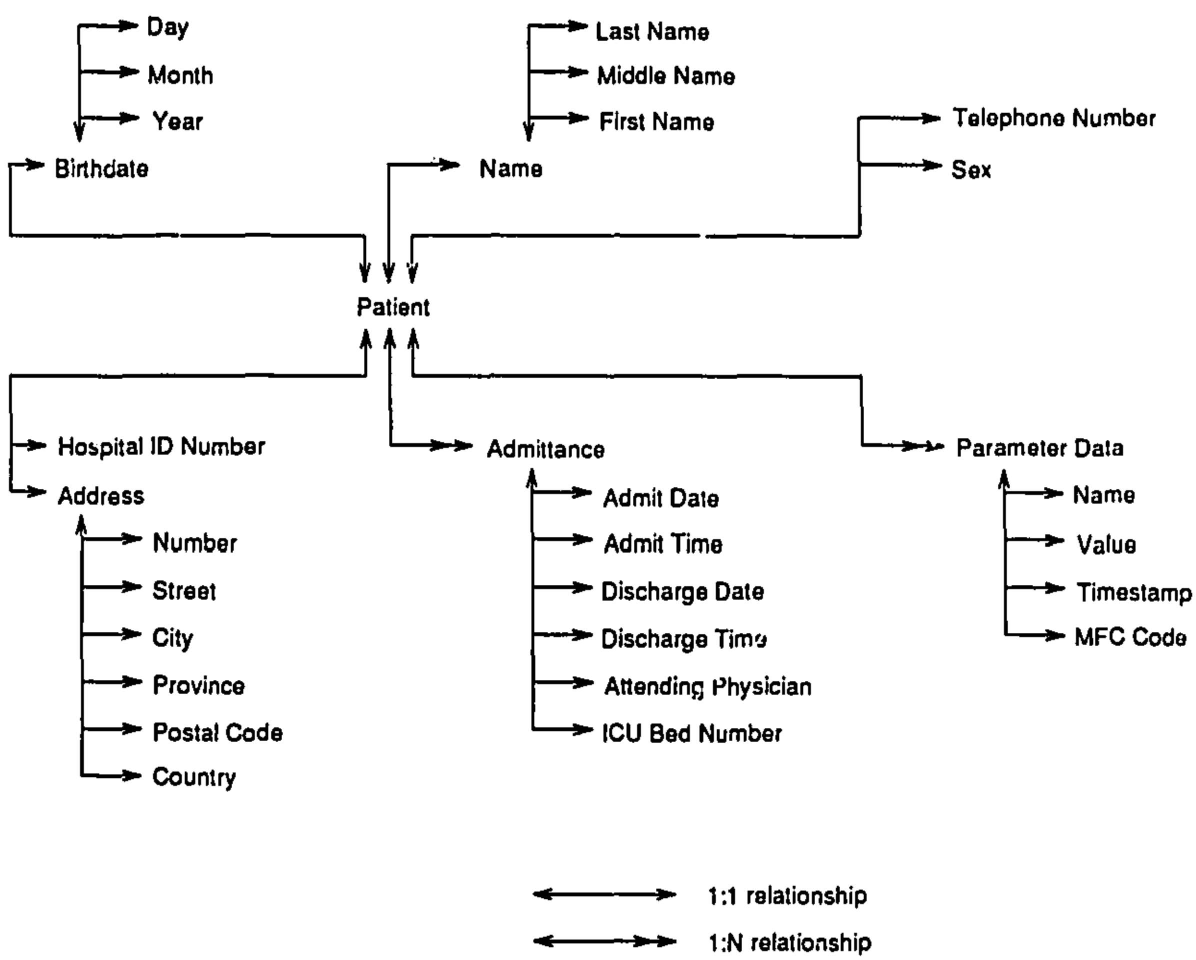


Figure 3.6: Entity-Relationship Diagram of PDMS database

The ADMITTANCES relation (Figure 3.8) will contain information about the different instances the patient was admitted to the ICU. Again the hospital ID is used as the primary key together with the admittance date and time. Together, these three values will represent an instance of a patient being admitted to the ICU. The admit and discharge date and times are stored as data types of system date and time, which are comparable to character strings, but allow for storage advantages. The Registration module will manage the data in the ADMIN_DATA and ADMITTANCES relations.

For the patient-specific data, several patient-specific relations are created, with each patient having his or her own set of tables. The relations, TSPARAM_*, SPARAM_*, MPARAM_* and HPARAM_* will store temporary second data, permanent second data, minute data, and half-hour data for a patient with a hospital ID of *. The definition of all these four tables is identical, with each tuple entry of these PARAM tables containing the parameter mfc code, associated time stamp and parameter data value. The mfc code is stored as a small integer (2 bytes), while the timestamp is a combined system date and time data type. The parameter value attribute is defined as a non-textual, variable length byte array. The variable length is is necessary since the parameter value may be single or triple valued. The MPARAM_123456789 relation (Figure 3.9) is representative of the PARAM relations. The TPSPARAM_* relation is somewhat special in that it is a circular table, where new second data points override previously stored points in a circular manner.

Figures 3.10 through 3.12 show some sample data which will be stored in the relations. Note that only one of the parameter data relations is shown. In Figure 3.12 the parameter values do not appear, since they are non-alphanumeric in nature. The actual Database Manager SQL definitions for the PDMS database are described in chapter 5.

With this database implementation, the different PDMS modules can now interact with the Database Manager in order to update and retrieve patient data. As

Query Manager for PDMS1990

Actions Constraints Exit Help

Table ADMIN_DATA

Select Actions above to add additional columns at the end of this table definition.

Column Name	Type	Length	Attributes
ID	Character (Fixed)	9	Data required, Text
FNAME	Character (Variable)	80	Text
MNAME	Character (Variable)	80	Text
LNAME	Character (Variable)	80	Text
SEX	Character (Fixed)	1	Text
BIRTHDATE	Character (Fixed)	10	Text
ADDRESS	Character (Variable)	80	Text
CITY	Character (Variable)	80	Text
PROVINCE	Character (Variable)	80	Text
POSTAL_CODE	Character (Variable)	80	Text
TELEPHONE	Character (Fixed)	13	Text

Figure 3.7: ADMIN_DATA Database Relation

Query Manager for PDMS1990

Actions Constraints Exit Help

Table ADMITTANCES

Select Actions above to add additional columns at the end of this table definition.

Column Name	Type	Length	Attributes
ID	Character (Fixed)	9	Data required, Text
DATE_IN	Date		Data required
TIME_IN	Time		Data required
DATE_OUT	Date		
TIME_OUT	Time		
DOCTOR	Character (Variable)	80	Text
BED_NUMBER	Small Integer		

Figure 3.8: ADMITTANCES Database Relation

Query Manager for PDMS1990

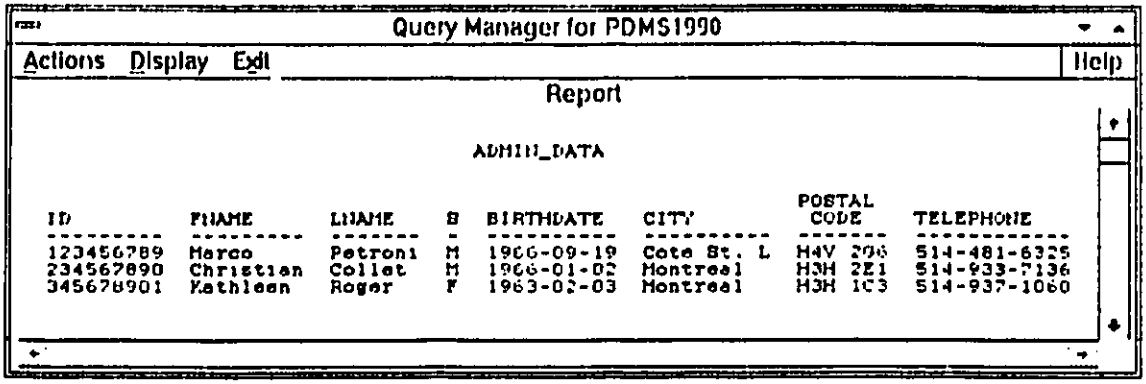
Actions Constraints Exit Help

Table MPARAM_123456789

Select Actions above to add additional columns at the end of this table definition.

Column Name	Type	Length	Attributes
MFC	Small Integer		Data required
TIME_STAMP	System Date and Time		Data required
VALUE	Character (Variable)	6	

Figure 3.9: MPARAM_123456789 Database Relation



Query Manager for PDMS1990

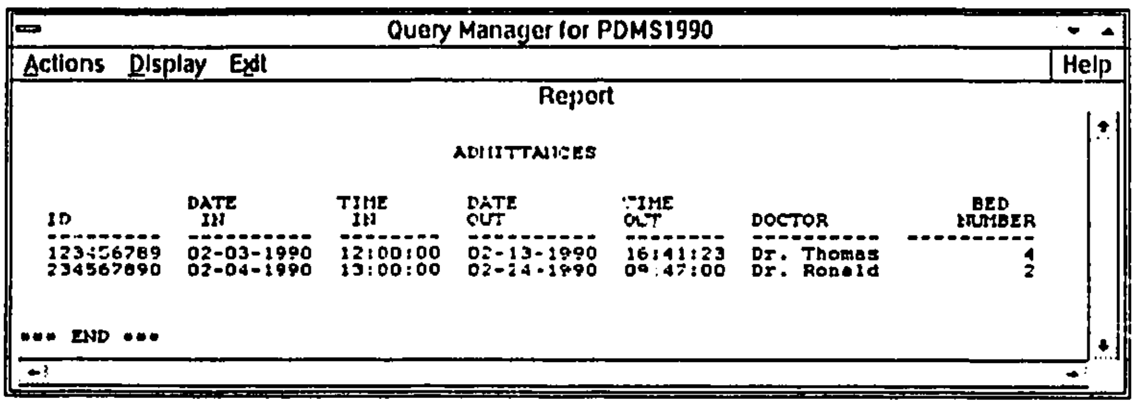
Actions Display Exit Help

Report

ADMIN_DATA

ID	FNAME	LNAME	S	BIRTHDATE	CITY	POSTAL CODE	TELEPHONE
123456789	Marco	Petroni	M	1966-09-19	Cote St. L	H4V 2Y6	514-481-6325
234567890	Christian	Collet	M	1966-01-02	Montreal	H3H 2E1	514-933-7136
345678901	Kathleen	Roger	F	1963-02-03	Montreal	H3H 1C3	514-937-1060

Figure 3.10: Sample data for ADMIN_DATA relation



Query Manager for PDMS1990

Actions Display Exit Help

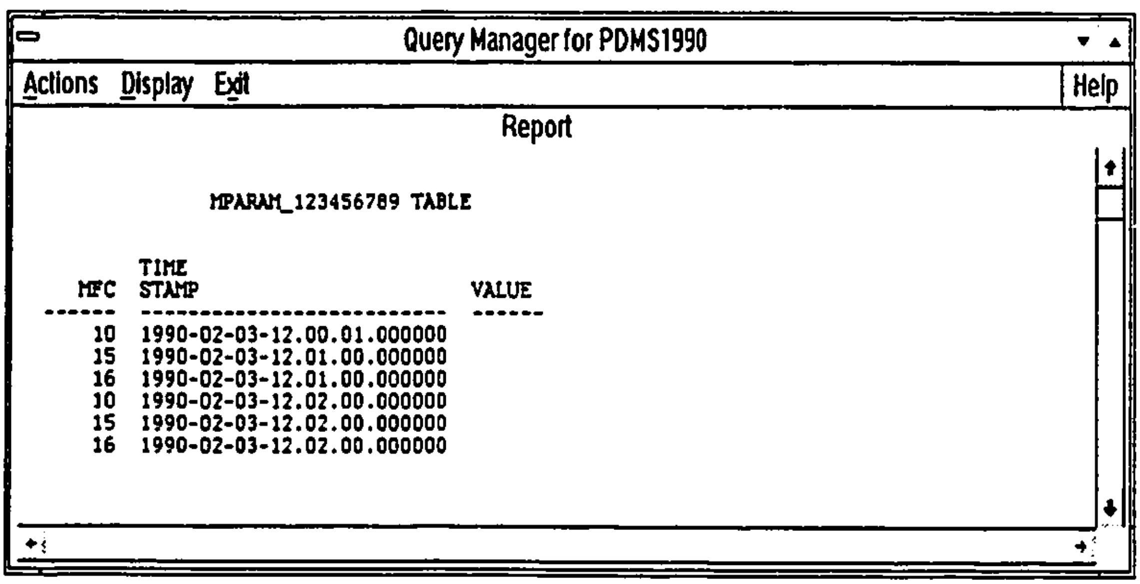
Report

ADMITTANCES

ID	DATE IN	TIME IN	DATE OUT	TIME OUT	DOCTOR	BED NUMBER
123456789	02-03-1990	12:00:00	02-13-1990	16:41:23	Dr. Thomas	4
234567890	02-04-1990	13:00:00	02-24-1990	09:47:00	Dr. Ronald	2

*** END ***

Figure 3.11: Sample data for ADMITTANCES relation



Query Manager for PDMS1990

Actions Display Exit Help

Report

MPARAM_123456789 TABLE

MFC	TIME STAMP	VALUE
10	1990-02-03-12.00.01.000000	
15	1990-02-03-12.01.00.000000	
16	1990-02-03-12.01.00.000000	
10	1990-02-03-12.02.00.000000	
15	1990-02-03-12.02.00.000000	
16	1990-02-03-12.02.00.000000	

Figure 3.12: Sample data for MPARAM_123456789 relation

the need arrives for even more patient data to be stored, then the database can be expanded easily by introducing new relations. Chapter 5 discusses the implementation of relations for storing wave data, alarm data and diagnostic codes.

Data Retrieval

Data retrieval from the PDMS is greatly simplified. Since the data is now easily accessible in the PDMS database, the stored PDMS data can be used by both PDMS modules and external applications. Modules requesting data need simply post the appropriate SQL query to the Database Manager. Special treatment is necessary only for the retrieval of parameter data by the Trend Display module. If the query requests data less than five minutes old, part of the parameter data must be retrieved from the memory resident queues and part from the relational database. This is obtained by splitting the data requests into two separate requests: one to the DLC module, the other to the Database Manager.

Another item which requires special attention is the display of 'real-time' parameter data by the Trend Display module. In this case, the Trend Display will present a self-updating screen of vital trends. To achieve this, the Trend Display module initiates a Dynamic Data Exchange (DDE) transaction with the DLC module. Dynamic data exchange is an information transaction based on the client-server model, which allows two or more applications to share data in an automatic fashion, according to a well defined protocol [Southerton, 1989]. Rather than having to continuously query the server for data, the client (Trend Display module) can request the server (DLC module) to transmit data automatically as soon as it becomes available, thus decreasing communication overhead.

With this new database implementation this PDMS data now becomes readily available and the medical staff may envisage other interactions with the patient data. This is further assisted by the presence of the Query Manager. Through the customized interface application-generator 4GL, customized query applications, with menus, push buttons, and customized forms, are foreseen. Such modules can be easily added to the present PDMS system or as separate programs accessing the database through the local area network. This approach will relieve the medical staff from having to learn any programming languages, and facilitates access to the data.

Data Security and Integrity

The new database design based on the OS/2 Database Manager solves many of the security problems of the previous implementations. For security of the data, the OS/2 Database Manager offers user screening and password protection. The first level of security is screening of individual users using the PDMS. This is possible since the Database Manager is integrated with OS/2's User Profile Management Services, requiring each database user to have an authorization ID, with password. Access priviledges to the database can then be controlled (through SQL 'GRANT' and 'REVOKE' statements) for different users. For medical staff using the PDMS, this would require that the nurse or physician log in (via the User Profile Management) before activating any PDMS module. The PDMS modules would then operate under the current user's authorization ID, and operate only on the data which is accessible.

The password protection feature involves assigning a password to the database objects as a whole. This feature is somewhat more restrictive, as it allows or denies access to the PDMS database as a whole.

The integrity issues have also been simplified with the new database implementation. While previous PDMS designs required two systems running in parallel to ensure fault-tolerant operation, this is no longer the case. The two systems, both connected to the CarePort interface, will be networked via a local area network. One system can be a dedicated server, with the other system being used for the PDMS interactive expert system or other external applications. Additionally, the two systems verify (via network diagnostic messages) that the other is functional. Should the server system fail, the second system can take over the task of patient data acquisition. Thus, although two systems are still required, the 'second' system is available for other uses while the server is functional. Furthermore, the two systems need not be located close together.

3.3.3 Data Archiving

To reduce the table number and storage space demand on the primary hard disk, the entire collection of tables in the PDMS database can be periodically transferred to mass storage. Traditional archiving devices such as tape will still be required for long term, off site archive storage. For intermediate mass storage, optical disks provide a storage medium which allows the data to remain on-line. This optical disk archive will increase the availability of the PDMS data. Transferring the database data to the optical disk can be performed on-line using the Database Manager BACKUP and EXPORT utilities. Only those tables and data belonging to discharged patients will be archived, and then deleted from the active PDMS database on the hard disk. Current ICU patients' data remains untouched. The time interval between archives will be determined empirically, taking into account storage space, data availability and accessibility requirements. For now, using a 630 Mbyte optical disk, a period of one year is envisioned. The archived collections can then be separate databases, labeled PDMSXXXX, where 'XXXX' would be the year (e.g. PDMS1990).

An appropriate media for this database archive would be the Write Once Read Many (WORM) optical disk. This would suit the large capacity required and preserve the legal requirements for the data. The WORM optical disk appears as an ideal storage device in a medical environment where non-alterable records have to be kept for legal purposes. This highlights one of the major differences of the PDMS database as compared to conventional database systems; deletions and updates are for the most part prohibited. Data simply accumulates and is stored for reference. Thus, issues such as the compaction of "holes" in the database are not relevant.

Storage requirements for the PDMS database will vary depending on what data is stored. With both minute and half-hour data being stored, in addition to patient administrative and admittance data, the storage requirements reach several hundred megabytes per year. If the stored parameter data is limited only to half-hour data, storage requirements can be reduced by as much as 50%. With optical disk systems having storage capacities ranging from 300 Mbytes to several gigabytes, storage space is not an immediate problem. This will change, however, when other PDMS data (alarm data, fluid balance data, waves) is stored. Careful evaluation of storage space will be required, with particular attention being paid to parameter data. Storage space requirements may dictate that minute parameter data not be stored, except for certain specified intervals, as is the case with second data. These issues are further discussed in chapter 5.

Retrieval of this data from optical disk storage is not a performance bottleneck. Most optical systems have performance ratings of 80 ms average seek time and 600 Kbytes/sec data transfer rates, making them only somewhat slower than hard disk storage. Furthermore, these optical disks can then be grouped in juke-box systems, and the various disks (anywhere from six to thirty) can still remain on-line for database queries and data retrieval. Parkin [Parkin *et al.*, 1990] has implemented such an archiving system and found it much more useful than stored paper records.

The optical disc system completes the multi-level storage structure (Figure 3.1), offering a long-term archiving solution which still allows some of the archived data to remain on-line. This can greatly facilitate diagnostic expert systems as well as aid research purposes.

Chapter 4 The PDMS/CareNet Simulator

This chapter describes the PDMS/CareNet simulator which was developed for testing of the PDMS software. The PDMS operates on patient data, much of it acquired electronically from the bedside monitors. At the research lab, away from the physiological monitor network, this data is not available, resulting in an inadequate testing environment. To remedy this, a simulator was developed to generate the patient data acquired from the network, and to simulate the functionality of the CarePort interface to the bedside monitor network.

Knight [Knight, 1990] stresses the importance of software reliability in medical systems. In particular, he is concerned with the dependability of system software for safety-critical systems — systems which have the property that human life could be endangered should the computer system fail. Three levels of dependability are examined: reliability, availability and safety.

Lal-Gabe [Lal-Gabe, 1990] reports on the importance of hazards analysis to identify failures inherent in software systems. Hazard analysis should be performed at early stages of product development to identify and to prevent any single fault from causing a dangerous failure in later stages of design. The results of hazard analysis are shown to be useful in determining the completeness of software testing and limiting the number and the amount of tests.

Proper testing of the software design requires that the software system be analyzed in an environment where the various components are all put to use. Schmidt [Schmidt *et al.*, 1989] suggests simulation as a means of testing and verifying software during its development. A simulated environment is more flexible than its real-world counterpart, and allows for almost any condition to be modeled. Pollacia [Pollacia, 1989] cites some of the advantages of simulation:

- Simulation allows experimentation without disturbing the real world system.
- Simulation provides estimations of the performance of a system for new parameters or operating environments.
- Alternative proposed designs can be compared and evaluated.

as well as some of its disadvantages:

- Simulation models may not be valid representations of the real world environment.
- Simulation runs may be lengthy and expensive in terms of both human and computer resources.

Studies by O'Neil and Gloroinski [O'Neil and Glowinski, 1990] and Fieschi [Fieschi, 1990] have shown the importance of proper validation and verification of medical software systems. The PDMS simulator is thus more than a debugging aid. Its usefulness for proper performance analysis and verification of the PDMS software will be invaluable.

4.1 The Hardware Environment

The simulator runs on an IBM PS/2 computer running the OS/2 operating system. The simulator can function on its own computer or it may share the same computer as the PDMS. Since the simulator and the PDMS communicate over an RS-232 serial link, both the simulator and the PDMS can run concurrently on the same PS/2. This is achieved by letting the PDMS use the COM1 serial port while the simulator uses the COM2 serial port. The two ports are then linked with an RS-232 cable. The Presentation Manager environment and the OS/2 screen management allow switching between the simulator display and the PDMS display. To evaluate the real-time performance aspects, enhanced simulator and PDMS versions should be tested using separate computers. One system will be dedicated exclusively to running the simulator in order to avoid interference with the PDMS software. This setup would also be useful when using the simulator as a training or teaching tool, since it would keep the trainee unaware of the actions being scheduled by the instructor.

In order to simulate the CareNet environment, the simulator must both generate the patient data acquired by the physiological bedside monitors and simulate the functionality of the CarePort interface. For this reason, communication between the simulator and the PDMS occurs through a serial line as though the PDMS were connected to the CarePort interface.

4.1.1 The CarePort Interface

The HP 78580A CarePort is a programmable interface between the network controller and the host computer, converting the proprietary format signals of the network to standard RS-232C messages which can be understood and decoded by the host. Most of the data acquisition activity of the PDMS deals with communicating with the CarePort interface. This communication process involves three different levels of activity: 1) a subset of the ANSI x3.28 - 1976 communication protocol must be respected for physical message transmission, 2) message information must be encoded or decoded from the logical message format of CarePort and 3) logical sources must be defined and manipulated correctly to obtain the desired network information.

Message Transmission

Communication between the host computer and CarePort involves message transmission across the serial link. An Enquiry-Acknowledge handshaking protocol is used to gain control of the serial communications link before message transmission. There are only two types of messages: requests and responses. A request is always sent by the host computer to CarePort which always answers back by a response message. However, this response might not be the next message sent by CarePort or it might not even be limited to one message.

Valid message requests can be grouped into three categories: unit messages, data messages and logical source messages. Unit messages deal with opening a communication link with the CarePort unit, configuring the data link, and resetting the unit. Data messages are those carrying the data coming from the physiological bedside monitors. Logical source messages instruct CarePort on what network data should be transmitted to the host computer.

Logical and Physical Messages

In sending a message to CarePort, two considerations must be kept in mind: what is the message and how to make sure the message is correctly communicated. The location and specification of the bits and bytes used in the message determine its significance. This is the logical message representing the command and its options. To make sure the communication is successful, additional control characters are added to the logical message before being transmitted on the physical communication line. This modified message is the physical message.

Physical messages are constructed according to a subset of the ANSI x3.28 - 1976 protocol. A physical message is made up of a logical message sandwiched between two series of control characters. An additional byte terminates the physical message, the Block Check Character (BCC), used for error detection. A detailed listing of CarePort messages and their corresponding logical message formats appears in the CarePort Programmer's Reference Guide [Hew, 1986].

Logical Sources

Logical sources are virtual connections between different data sources designed both to increase the efficiency of the serial communication line and to make the host-network interface easier to program. For example, a logical source can be specified for a particular association of beds or for a list of parameters to be recorded, or both. Up to ten different logical sources can exist at any time, identified by a number from 0 to 15. The host computer can thus define and manipulate the desired options for a group of data sources by referring to a single logical source number (LSN); referring to a logical source by its LSN number also decreases the length of the transmitted message. Admission, suspension or discharge of patients on the network is done through the use of logical source definitions.

Logical sources can be in three states: not connected, connected but suspended, and connected and active. Connecting a logical source is equivalent to defining it with its options to CarePort. Activation signifies that CarePort will give the available data corresponding to the logical source definition. Suspension is the opposite to activation: the logical source remains connected but no data will be transmitted.

Acquisition Methods

Once a logical source has been defined (connected) and activated, three mechanisms exist to acquire the data.

1. Polled Mode. In polled mode, CarePort will send the most recent data only when the corresponding request for data has been issued by the host computer. Polling may not be performed more frequently than every 1024 ms.

2. Scheduled Automatic Mode. In this mode, the host computer specifies how often (frequency) and for how long (duration) CarePort should automatically

transmit data.

3. Continuous Automatic Mode. Somewhat similar to scheduled automatic mode, this mode instructs CarePort to transmit logical source data every 1024 ms for the duration of activation of the logical source. In this mode CarePort also provides buffering of the data.

4.2 Simulator Modules

The simulator uses OS/2's Presentation Manager window environment for building its user interface, while the multitasking abilities of OS/2 are used to modularize the simulator design. The simulator is essentially divided into two modules:

1. The Network Module which emulates the CarePort interface and communicates with the PDMS. This module receives and responds to PDMS requests, and performs the actual data transmission.
2. The Data Module which is responsible for what patient data is being synthetically generated and how. The type of data and method of data generation is specified by the user and can be varied for each patient.

The simulator begins by displaying its main window, Figure 4.1, where fourteen beds are displayed using an icon and accompanying text to label the beds. The state of the bed icons (highlighted or not) indicates whether or not the simulator is generating any patient data for that bed. This allows the user to get the state of the simulator at a glance. The pull-down menu items offer a variety of choices:

- Open the Network Module window.
- Open the Data Module window.
- Display queued simulator events.

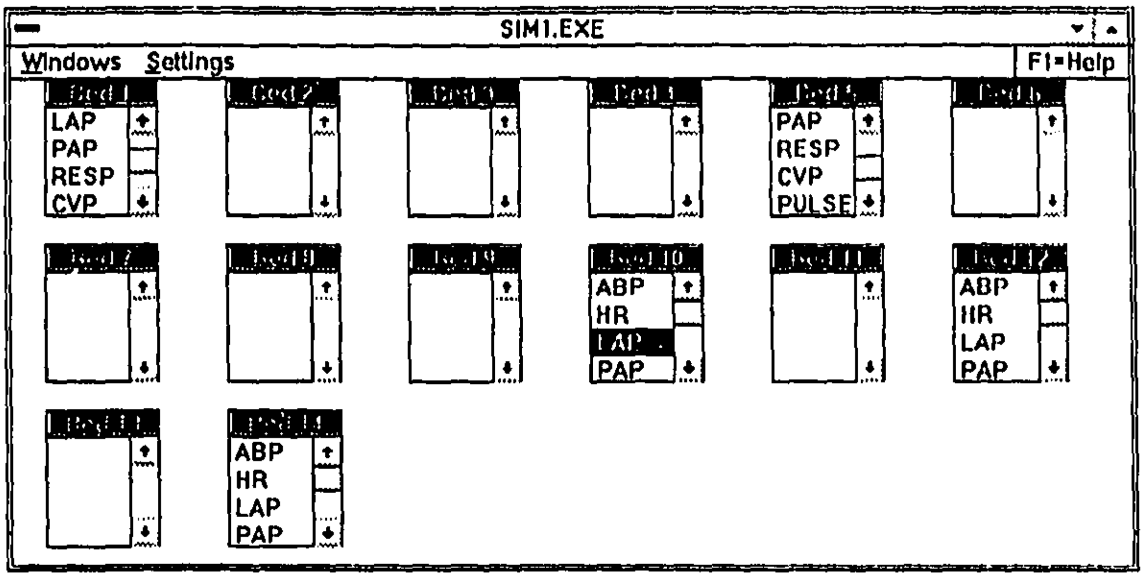


Figure 4.1: Main Simulator Window

- Adjust display settings.
- Help on simulator functions.

These menu items can be selected any time from the main window's menu bar. The help function is also available on other windows and provides case sensitive on-line help. The setting item allows users to personalize some of the display settings such as the color used for windows, menus and text. Screen design is also an important aspect, and for this the Presentation Manager environment permits users to resize, move and overlap windows.

By selecting the appropriate items in the *Windows* menu, the Network Module window, the Data Module window, or the Events window can be opened. Figure 4.2 shows the menu tree for the simulator modules, indicating the various operations available at the different levels. The various simulator displays are accessible at any time via the *Windows* menu of the main simulator display, or they may be displayed simultaneously.

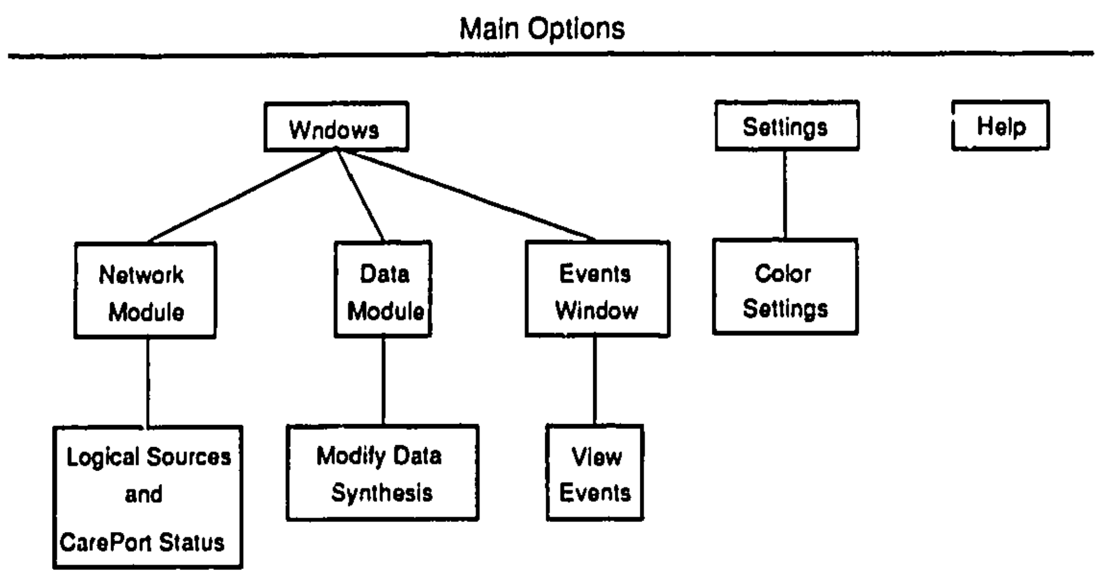


Figure 4.2: Simulator Menu Tree'

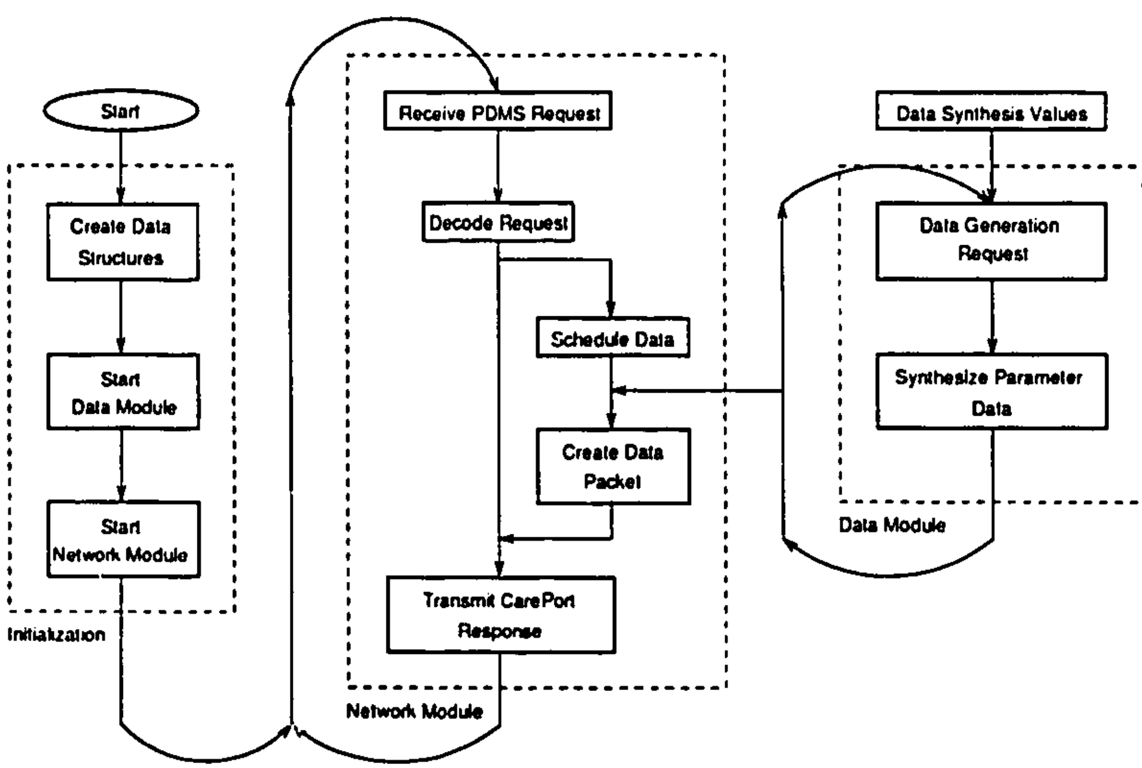


Figure 4.3: Overall Flow Diagram of the Simulator Program

4.2.1 The Network Module

The Network Module is responsible for simulating the CarePort interface. This module must accept commands from the PDMS and respond to them using the proper communications format. The various messages which must be simulated and subsequently transmitted to the PDMS include unit messages, data messages and logical source messages. Handling these message requests and responses is straight forward, except for the data messages which require data transmission. This data will be generated by the Data Module and must be transmitted to the PDMS at periodic intervals.

These features can easily be implemented by exploiting OS/2's multitasking features. The Network Module is implemented using multiple threads. One thread remains in a perpetual loop of receiving, decoding and replying to requests sent out by the PDMS, while the other thread handles the transmission of the data requested by the PDMS. The specified time interval is respected by employing OS/2 system timers.

Figure 4.4 shows the CarePort window of the Network module display. Here the user may view the state of the logical sources, the current communication link parameters, and the simulated status of CarePort and the bedside monitors. Selecting one of the logical sources in the *Logical Sources* listbox will display in the *Specifications* window all the beds currently defined for that logical source. Selecting one of these beds will then display the currently defined data items for that bed. Since none of these items is simulator generated or specified, they are not open to user modification.

CarePort messages about status of CarePort and bedside monitors are generated by the simulator, and can be varied by the user. To this end, the user may specify various error conditions such as overrun or unit test errors by selecting the *Add Error* push-button. Simulation of these errors is important in verifying proper PDMS functionality during error conditions.

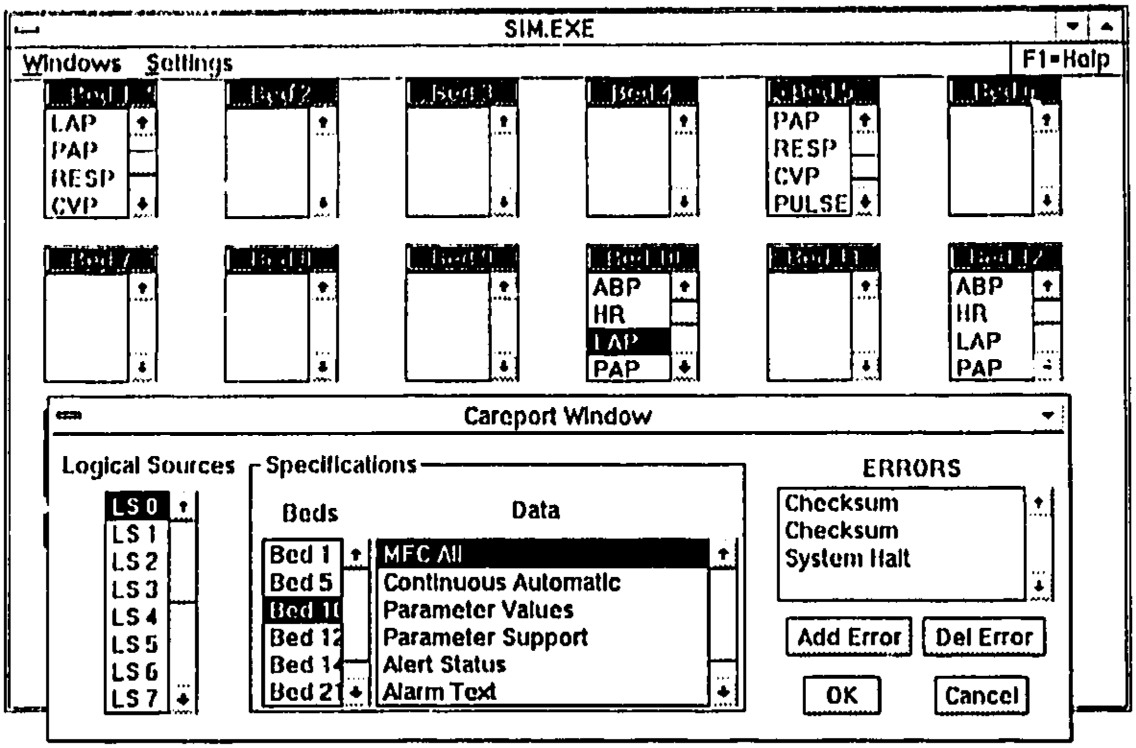


Figure 4.4: Network Module Window

4.2.2 The Data Module

The Data module replaces the network of bedside monitors and generates synthetic patient data for transmission to the PDMS. This requires generating the various data types available from the bedside monitors: parameter data, wave data, alarm data, support text and inop text. Of these, only parameter and wave data involve actual generated data values. Alarm data such as threshold alarms or inop alarms are dependant on the value of the corresponding parameters, as are support and inop text. The present implementation generates only parameter data and corresponding alarms.

The current data synthesis algorithm uses a random distribution centered about a specified mean value and with a specified variance. The variance value specified by the user ensures that the data value is kept within specified bounds. The random distribution results in data values which vary, simulating typical patient trends.

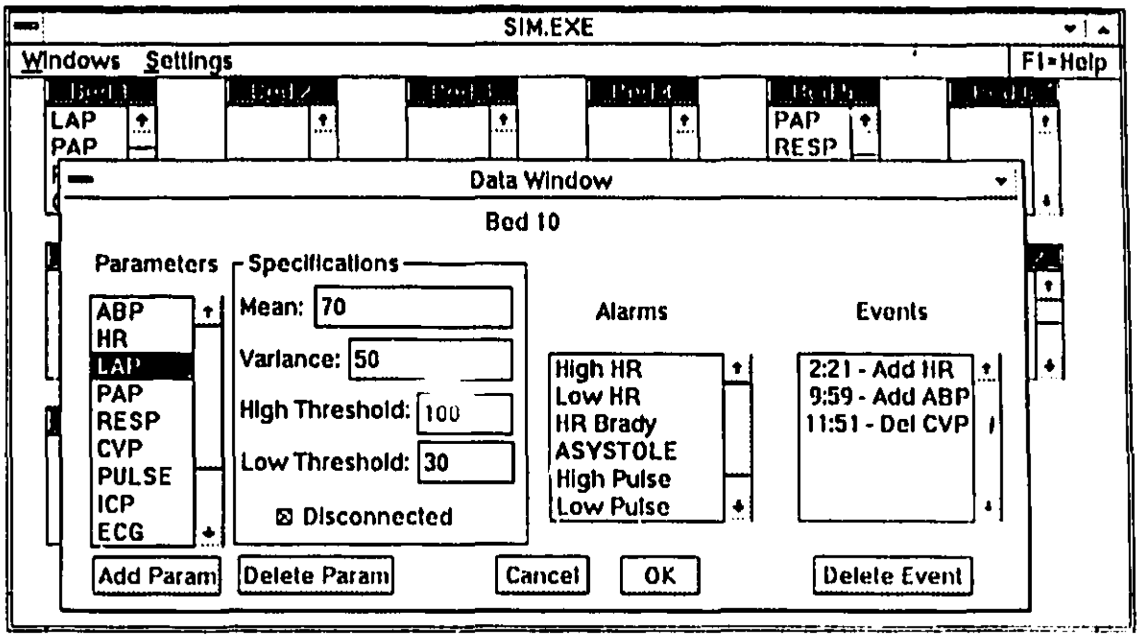


Figure 4.5: Data Module Window

A constant value can also be generated by specifying a variance of zero. This algorithm is somewhat simplistic but nonetheless effective. Future extensions may incorporate more elaborate algorithms along with the ability to read parameter data from a file.

New parameter data values are calculated every 1024 ms, to match the data rate provided by the bedside monitors. The calculation and update is performed by a dedicated, timer activated thread which accesses the global data structures and and performs the updates. Although the PDMS may not have requested periodic second data, generation of the synthetic data every second is implemented to simulate the stand-alone capability of the bedside monitor unit. Additionally, for those parameters which have threshold limits or disconnect states, the thread determines if any alarms must be generated. Threshold alarms are generated when the parameter value exceeds the specified high threshold or falls below the low threshold. Disconnect alarms are generated when the disconnect flag is set. The threshold values and disconnect status are all specified by the user of the simulator program.

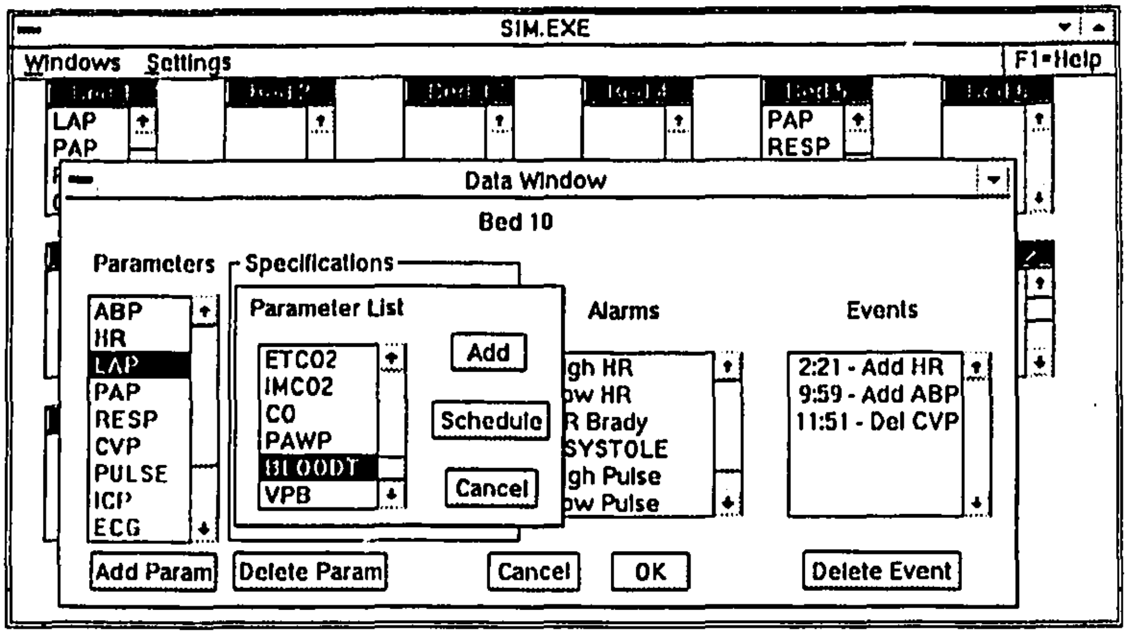


Figure 4.6: Data Module Window with Add Parameter Dialog Box

The Data module window, Figure 4.5, displays the patient data being generated, if any, and the current generation scheme being used for the selected bed. If the user wishes to modify the parameter types being generated, the *Add Param* button is selected and a dialog box (Figure 4.6) is opened. Through the dialog box, parameter addition, deletion and modification may be performed. Modification of a parameter value is performed by modifying the values used in the generation scheme (*Specifications* entry boxes, Figure 4.6). After a parameter addition, deletion or modification has been specified, the user has a choice between letting the action take effect immediately or scheduling it to occur at a specific time. If scheduled mode is chosen, the user is prompted for the time at which the event will occur. The event is then queued in the event queue, until which time it will take effect. This feature allows several parameter modifications to take effect simultaneously by scheduling them to occur at the same time. The list of queued events can be viewed in the *Events* window (Figure 4.5). The queued events can also be viewed by selecting the *Events* item in the main window's *Windows* menu.

Chapter 5 Implementation, Results, and Future Extensions

This chapter describes the implementation issues and status of the database and simulator. Sample results, showing the operation of the simulator and database are then presented and reviewed to determine possible shortcomings and improvements. Lastly, future extensions to the database architecture, simulator and to other PDMS modules are presented.

5.1 Implementation

Both the database and simulator were implemented in the C programming language and built using the Microsoft C compiler version 6.00. All function definitions comply with the ANSI C standard. The database, simulator and other PDMS modules were successfully ported to OS/2 Extended Edition version 1.3.

5.1.1 Relational Database Implementation

The PDMS database was implemented using the relational database management services available from the OS/2 Extended Edition Database Manager. Interaction with the Database Manager was performed via embedded SQL statements, in the C source code. Embedded SQL serves as a database programming language. It contains the same functionality as interactive SQL, but with additional statements whose purpose is to provide the support required by programming languages in order to interface with the Database Manager.

The embedded SQL statements used to create the various database base tables are shown in Figure 5.1. The parameter value tables are specific to each patient. Uniqueness is ensured by using the patient's hospital ID as part of the parameter table's name. As shown in Figure 5.1, when creating the parameter tables, the patient's hospital ID number is retrieved from the *id* variable.

Note the declaration of the *value* attribute for the PARAM relations as a VARCHAR FOR BIT DATA data type. This data type allows the storing of data which may not have a constant length. The FOR BIT DATA data type is treated as a structure with two fields: a length indicator, and a byte array which holds the values. This is needed since the parameter data values may be two or six bytes in length, depending on whether the parameter is single or triple valued. By using a variable length value attribute, regularity of the PARAM table is maintained. A VARCHAR for text data type could not have been used in this case since a null value could not have been stored, due to its use as a character string terminator.

The ADMIN_DATA and ADMITTANCES tables need only be created once, at the creation of the PDMS database. The PARAM tables, however, will be created for each patient admitted to the PDMS. When a new patient is admitted to the PDMS, the Patient Registration module will update the database with the new admittance info, and if needed, a new set of PARAM tables will be created.

For the parameter data, embedded SQL statements in the Trend Display module are responsible for retrieving the requested patient data from the database. The SQL statements will retrieve the data for the appropriate patient and for the specified time interval constraints.

The insertion of parameter data into the PARAM tables is performed by the DLC module. The parameter data is obtained from the circular data queues and written to the relational database. This is performed at a periodic interval of five minutes. Flushing the data queues at a periodic interval reduces the disk write overhead by transferring the data in larger packets.

ADMIN_DATA TABLE

```
EXEC SQL
CREATE TABLE ADMIN_DATA
(ID            CHAR(9)        NOT NULL TEXT,
FNAME          VARCHAR(80)    TEXT,
INITIALS       VARCHAR(80)    TEXT,
LNAME          VARCHAR(80)    TEXT,
SEX            CHAR(1)        TEXT,
BIRTHDATE      CHAR(10)       TEXT,
ADDRESS        VARCHAR(80)    TEXT,
CITY           VARCHAR(80)    TEXT,
PROVINCE       VARCHAR(80)    TEXT,
POSTAL_CODE    VARCHAR(80)    TEXT,
TELEPHONE      CHAR(13)       TEXT,
PRIMARY KEY (ID));
```

ADMITTANCES TABLE

```
EXEC SQL
CREATE TABLE ADMITTANCES
(ID            CHAR(9)        NOT NULL TEXT,
DATE_IN        DATE           NOT NULL,
TIME_IN        TIME           NOT NULL,
DATE_OUT       DATE,
TIME_OUT       TIME,
DOCTOR         VARCHAR(80)    TEXT,
BED_NUMBER     SMALLINT,
PRIMARY KEY (ID, DATE_IN, TIME_IN));
```

MPARAM_* TABLE

```
strcpy(sql_stmt,"CREATE TABLE ");
strcat(sql_stmt,"MPARAM_");
strcat(sql_stmt,id);
strcat(sql_stmt," (MFC SMALLINT NOT NULL, TIME_STAMP TIMESTAMP NOT NULL,");
strcat(sql_stmt,"VALUE VARCHAR(6) FOR BIT DATA, PRIMARY KEY (MFC,TIME_STAMP))");
EXEC SQL
EXECUTE IMMEDIATE :sql_stmt;
```

Figure 5.1: SQL Statements To Create Database Base Tables

5.1.2 Simulator Implementation

The simulator is currently implemented as a command line utility with no user interface [Fumai *et al.*, 1987]. Command line parameters may be specified for certain options (such as which serial port to use), but once started, the Simulator will run in the background, with no display and no user interaction. Although lacking user interaction, the CarePort interface emulation and synthetic parameter generation functionality is present. The source code for the Simulator is divided into three program files totaling about 4000 lines of code. The size of the executable file is approximately 100 Kbytes.

The proposed user interface presented here will be merged with the existing simulator program, allowing a user to vary parameter values and other settings, as described. The new user interface will use the Presentation Manager graphical user interface environment of the OS/2 operating system, which adheres to the Systems Application Architecture (SAA) definition for user interfaces. The text and graphical drawing operations in the various Simulator windows rely on the standard Presentation Manager API functions available from the Presentation Manager libraries.

5.2 Sample Database Results and Evaluation

This section presents some benchmark results obtained while operating the PDMS in the development lab at McGill. The sample data was generated using the PDMS/CareNet Simulator. Performance evaluations of the OS/2 Database Manager, using loading, indexing, retrieval and select operations derived from the AS^3AP (ANSI SQL Standard Scalable and Portable) benchmark tests for relational database systems, are readily available in the microcomputer literature [Venditto, 1991]. The objective, therefore, was not to benchmark the performance of the OS/2 Database Manager, but rather to evaluate the specific PDMS database

Transaction	1 Bed	7 Beds	14 Beds
Admit Patient (new ID)	2000ms	2010ms	2010ms
Admit Patient (old ID)	990ms	995ms	996ms
Periodic Data Write (new ID)	150ms	500ms	2200ms
Periodic Data Write (old ID)	155ms	520ms	2220ms
Minute Data Retrieval (new ID - single trend)	500ms	505ms	500ms
Minute Data Retrieval (old ID - single trend)	600ms	600ms	605ms
Minute Data Retrieval (new ID - five trends)	2000ms	2000ms	2000ms
Minute Data Retrieval (old ID - five trends)	2200ms	2105ms	2205ms
Half-Hour Data Retrieval (new ID - single trend)	490ms	500ms	490ms
Half-hour Data Retrieval (old ID - single trend)	580ms	550ms	590ms
Half-hour Data Retrieval (new ID - five trends)	2050ms	2040ms	2060ms
Half-hour Data Retrieval (old ID - five trends)	2150ms	2160ms	2155ms

Table 5.1: Sample Database Transaction Benchmarks

implementation and determine if this new implementation would perform within acceptable limits.

Table 5.1 shows the response time of various transactions involving the database. These response times were measured on a PS/2 model 80 computer, with an 80386 16 MHz microprocessor, 8 MB of RAM memory, 150 MB of hard disk storage (20 ms access time, 800 KB/sec data transfer rate), running version 1.3 of OS/2. The response times were calculated using OS/2 system timers to record the transaction duration to the nearest 10th millisecond. For the benchmarking, OS/2 was configured for protected mode only, with swapping and virtual memory features fully enabled.

The transaction response times are shown for cases where 1, 7 and 14 beds were registered with the PDMS. Observing the transaction response time under these three conditions allows a determination of the load effect on database performance. As well, all the transactions were performed for an old and new hospital ID. The old hospital ID will have previously been registered with the PDMS, and will have PARAM tables which already contain one week's worth of parameter data. This is contrasted with the new hospital ID, whose PARAM tables will be initially empty, but will require creation. The various transactions are described below.

1. <u>Admit Patient</u>

The Admit Patient transaction involves admitting and registering a new patient to the PDMS, via the Registration module. This will require updating the ADMIN_DATA and ADMITTANCES tables and the new ID will additionally require creation of the various PARAM tables.

The response times under the three test conditions show very little difference. The increase in response time for the 7 and 14 bed cases is most likely associated with the increased load on the system due to the increased PDMS/CarePort communication.

The admit patient response times are within acceptable limits. The admit new ID times are higher than the admit old ID times, as expected. This reflects the increased activity of creating the PARAM tables. Even with this activity (performed only once) the response times are acceptable.

2. <u>Periodic Data Write</u>

This transaction is performed by the DLC module when it periodically flushes the values in the circular data queues to the database. This will require inserting data rows in three relations: TSPARAM_*, SPARAM_* and MPARAM_*. Every sixth five minute interval, a new half-hour data point will be available and this will then require an update of the HPARAM_* table. To perform these updates requires the patient hospital ID number. Since this is stored in the memory resident patient table (Figure 3.2) no database access is required to retrieve this information.

The response times for writing the parameter data values to disk are all within acceptable limits. The increase in response times when going from 1 to 7 to 14 beds is expected since more patient data is being written. What remains to be determined is the exact trade-off point at which writing data in larger blocks does not provide any additional reduction of disc-write overhead. Once determined, this value will allow proper calibration of the data write interval, to accommodate both integrity and efficiency concerns.

The nearly identical response times for old and new ID indicate that this has little effect. This is most likely due to the fact that the periodic data writes will always append data to the end of the PARAM tables. This behaviour contrasts with that of the retrieval response times.

3. Data Retrieval

This transaction is performed by the Trend Display module when a user asks to review a patient's parameter data for a specified time interval. This will require retrieving parameter data values and their corresponding timestamps from one of the PARAM tables, depending on whether the user asked for second, minute, or half-hour data. The data retrievals all involved data requests spanning two screenfuls (i. e. 100 data points) although the size of the specified time interval is not a factor since the Trend Display module will always clip this data request into one for 50 data points (i.e. exactly one screenfull). The retrieval requests are further subdivided into requests for 1 and 5 trends. This will allow an analysis of the effect which multiple trend requests (the normal scenario) will have.

The response times for all categories may appear somewhat slow, but are nonetheless acceptable. One noticeable result is that the minute data retrieval times are nearly identical to the half-hour retrieval times. This was expected since the only discernible difference between retrieving fifty minute data points versus retrieving fifty half-hour data points is that the MPARAM table will contain more data, which accounts for the slight time variations.

Comparing single trend versus multiple trend retrieval, one notices a sizable increase in the retrieval time. This result was unexpected, at first. Further examination revealed that the data request function of the Trend Display module was initiating a separate database request for each different trend. The complete data retrieval transaction was thus being repeated for every different trend. This would explain the almost proportional increase. The multiple trend request response time is not, however, exactly five times the response time of the single trend request since the Database Manager will buffer (move into RAM) the table indexes and catalog during a request, making subsequent immediate requests on the same table execute faster.

Since the normal everyday operation of the PDMS will have multiple parameter data trends being displayed, it is important that the response time for a multiple trend data request be improved. To achieve this requires altering the data retrieval procedures of the Trend Display module. Currently, one screenful of data points is obtained by requesting data points for the specified time interval on a per trend basis. This results in an SQL statement of the form:

```
SELECT TIME_STAMP, VALUE FROM MPARAM_1234567ᶠ9
WHERE MFC = 'trend'
AND TIME_STAMP > '1990-02-03.12.00'
AND TIME_STAMP < '1990-02-03.12.51'
```

being repeated for each trend. To improve response times, the various trends can be grouped into one data request, which would take the form of:

```
SELECT MFC, TIME_STAMP, VALUE FROM MPARAM_123456789
WHERE TIME_STAMP > '1990-02-03.12.00'
AND TIME_STAMP < '1990-02-03.12.51'
```

Grouping all the trends for a specific time interval into one request will require a larger buffer for storing the data, but will result in a faster retrieval time. Once retrieved into a RAM buffer, the data can then be manipulated as desired.

Comparing the new ID response times versus those for an old ID shows that the the old ID retrieval times are consistently slower than the new ID times. These results show that the amount of data in the PARAM tables affects the retrieval times. The current database design makes use of this result in that a new set of patient-specific PARAM tables are created for each different patient admitted to the PDMS. The alternative approach would be to store the parameter data in bed-specific tables [Fumai *et al.*, 1991] which would result in a fewer number of tables being created, but with the drawback that the tables would grow to a considerable size. The large size would then adversely affect performance of data retrieval, as indicated by the response times in table 5.1.

5.3 Future Extensions

5.3.1 Database Extensions

Extended Data Support

The current database implementation only supports a patient's parameter data and administrative data. While this may be suitable for test and evaluation purposes, it is by no means sufficient for any full scale implementation. Consequently, the database design must be extended to include, at the very least, such data as waves and alarms from the bedside monitors, fluid balance data, diagnostic codes, and progress notes or observations.

The alarm and fluid balance data can be easily accommodated in separate tables using a format similar to the PARAM tables. An alternative approach might be to include the alarm data which relates to parameter data values, in the PARAM tables. Other alarm or inop data would then be placed in a separate table.

The diagnostic codes assigned to each patient could be integrated into the ADMITTANCES table, since there are only one set of codes per patient admittance. Notes and observations of the medical staff are more difficult to manage under the relational model, since these are free-text based data types. One possibility is to include the whole text as a very long VARCHAR data type. The drawback is that the text could not be searched by the database system but would require custom processing programs. An alternative that may prove to be feasible is to store short, two to three word comments associated with a particular keyword (e. g. Skin color: rosy, Fingertips: warm and saturated, Forehead: hot). If a standard set of keywords can be adopted, the text based description could be stored as values of these keywords. This, however, would require a standardization of the care process.

Another stumbling block for the relational model is the handling of wave data. This problem can in fact be extended to include all data of the *multimedia* type, such as video images, audio data and free text. The problem stems from the fact that a wave sample does not easily fit the row-column format of a relational table. This type of data has the property of requiring large storage space (e. g. storing digitized samples of CD-DA audio sound requires 10 Mbytes of storage space per minute). One possibility is that of storing this data as a BLOB (Binary Large Object) in the relational table [Shetler, 1990]. But this results in data which can not participate in a content search.

The most promising solution seems to be the use of an object based data model. The underlying object oriented database will store these data types as objects, thus combining the large binary data with procedures for interpreting it. A multimedia document, stored in such a manner, would allow content searching and retrieving not only on text, but on audio, video and image data as well [Meghini *et al.*, 1991]. The possibility of integrating an object based database into the PDMS is currently being investigated [Saab *et al.*, 1992].

Improved Data Management

Since the present PDMS is a prototype system, many areas of PDMS data management are still unimplemented. Improvements in these areas will result in a more complete data management system.

The PDMS is not linked to the Hospital Information System (HIS). Without a link to the HIS, the PDMS must store the patient administrative data, resulting in data duplication. It would be desirable if the PDMS were linked to the HIS, so that not only could this information be stored in one place only, but other data could also be obtained from the HIS and serviced for useful interpretation in the context of the intensive care unit.

Another possible enhancement is the direct interfacing of the infusion pumps, which have RS-232 ports in many cases. As for the physiological monitor data, acquiring the infusion pump data would relieve the medical staff from the data acquisition task while obtaining continuous data rather than single measurements at long intervals would improve the accuracy of the data. It could even be possible to implement a feedback loop for close-loop patient control, but this has many serious overtones, both medical and legal. The fluid flow from the infusion pumps could be controlled in real-time according to the patient data.

Such a system is currently unfeasible, from an economical viewpoint, because the ICU is equipped with a number of pumps from different manufacturers which implement different protocols. The required standardization, however, can be achieved and extended to other devices through the use of the emerging Medical Information Bus (MIB) standard [Figler and Stead, 1990].

Networking

Implementing a distributed database allows ease of access (data transparency) to the PDMS data. Further advantages can also be achieved by a network implementation which allows users to access the PDMS database from remote locations. Remote stations then catalog the PDMS database as a remote database, and subsequently access it as if it were a local database. The underlying network communications is handled by the Communication Manager's Remote Data Services component.

Of course in such a network environment, security of the PDMS data can become a serious concern. Current security control may not be sufficient. Investigation into more advanced security mechanisms [Jajodia and Sandhu, 1991] may be required.

The current version of Database Manager for OS/2 version 1.3 does not support fragmentation of the database. This implies that all database objects must be stored in a central location. In a networked environment, however, if the data is stored at a central location, the load on the central server can become significant as all bedside and remote terminals request patient data. One solution is to create a distributed database and make use of table, row and/or column fragmentation. This would allow various tables to be stored in different locations. As an example, all patient parameter data from a specific ICU bed would have its own dedicated storage server. Such a server would not have to be as performant as the central server, since it would only process requests for specific patient data. In such a scenario, perhaps the bed-specific PARAM tables would prove more useful than their patient-specific counterpart. This must be examined further when a version of the Database Manager which supports database fragmentation is available.

Although the PDMS was originally designed for use on a central station, the emergence of bedside terminals will require some restructuring of the PDMS design. A distributed database implementation can be based on a client-server architecture, with a central DLC module receiving requests from remote bedside units. Bedside terminals also provide an ideal platform for other productivity software such as nursing workload managers and patient care plans [Roger *et al.*, 1991]. This gives rise to a computer system which provides support for collaboration among various users and tasks.

5.3.2 Simulator Extensions

In order to follow the evolution of the PDMS software, the simulator will have to be modified by adding more specialized modules. One important area is improving parameter generation schemes by adding more elaborate parameter synthesis algorithms. This would not only provide valuable test data, but it will also assist in the development of an expert system shell for the PDMS. Also, the full range of network patient data, including wave data, support text and inop text, should be made available for synthesis.

As well, a welcomed enhancement would be the ability to read parameter data from a file. This would allow parameter data captured at the hospital to be used for tuning the PDMS software. Additionally parameter data from cronic conditions (heart arrest, lung.failure) could be stored as separate groups and used for test cases.

With such extensions, the use of the simulator as a training tool for novice PDMS users emerges. Generating synthesized patient data would allow the trainees to learn proper PDMS operation with a variety of patient data.

5.3.3 PDMS Extensions and Research Directions

User Interface

The present PDMS user interface is based largely on a menued user interface. Shneiderman [Shneiderman, 1987] suggests that a menu based interface can relieve much of the confusion and learning curve associated with command-line systems. The menuing system also reduces errors due to incorrect commands, frequently experienced by casual users. This menu hierarchy implementation is useful for a central station implementation of the PDMS.

On the other hand, a more flexible arrangement could prove to be more productive by minimizing such "menu navigation"; one way to implement this would be through the use of a windowing environment, such as the Presentation Manager. Each module could thus reside in its own window, which could be freely moved around and resized. The System Manager main menu, as well as other menus could be implemented as pull-down menus from a standard "menu-bar", while a drag and drop functionality can be implemented through the use of icons. Such direct manipulation interfaces, where objects on the screen can be pointed to and manipulated using a pointing device, relieve the menuing system from its restrictiveness. Components are presented visually and actions are performed by pointing, not typing. These visual, direct manipulation systems are now widely accepted.

Although such a window-icon-mouse paradigm has proven to be useful in helping new users master complex software systems, it is unclear whether it could really enhance the function of the PDMS. It could safely be assumed that the users of the PDMS would each receive adequate training, and thus could be taught to use a more traditional user interface based on function keys and menus effectively. Too much freedom and flexibility might cause more confusion than necessary. Furthermore, the large overhead generated by a windowing environment must be kept in mind, especially since response time is always a critical factor in the acceptance of any software system by its users.

A new user interface option currently being evaluated is the use of a speech interface, especially for such bedside tasks as fluid balance data entry. Petroni [Petroni *et al.*, 1991] describes a speech system which combines both speech recognition for data entry and speech generation for feedback and verification of commands. This interface will provide an "eyes-free" and "hands-free" alternative to current user interaction techniques.

Chapter 6 Conclusion

This thesis has presented the development of a database for an intensive care unit patient data management system. A literature survey of medical systems and a discussion of the advantages of a computer data management system was first presented. This was then followed by an overview of the Patient Data Management System (PDMS) being developed. The database design was then detailed and its current implementation presented. Sample results are used to indicate the database's suitability in the PDMS environment, and improvements to the current design are suggested.

This manuscript has also discussed the development of a simulator for the PDMS software. The simulator emulates the CarePort interface and generates synthetic patient data, facilitating PDMS testing. Synthetic patient data generation also provides the data required for database evaluation. The simulator functionality is described and the design of an enhanced user interface is presented.

References

[Abrami and Johnson, 1990] P. F. Abrami and J. E. Johnson, *Bringing Computers to the Hospital Bedside: an Emerging Technology*. New York, NY: Springer Publishing Company, 1990.

[Albisser, 1989] A. M. Albisser, "Intelligent instrumentation in diabetic management," *Crit Rev Biomed*, vol. 17, no. 1, pp. 1–24, 1989.

[Albrecht and Lieske, 1985] C. A. Albrecht and A. M. Lieske, "Automating patient care planning," *Nursing Management*, vol. 16, pp. 21–26, July 1985.

[Alesch *et al.*, 1991] F. Alesch, E. Donauer, and G. Heinen, "The use of personal computers for the collection, storage and analysis of data from neurosurgical intensive care monitoring," *Anaesthesist*, vol. 40, pp. 52–56, January 1991.

[Andreoli and Musser, 1985] K. Andreoli and L. A. Musser, "Computers in nursing care: The state of the art," *Nursing Outlook*, vol. 33, pp. 16–21, January-February 1985.

[Aukburg *et al.*, 1989] S. J. Aukburg, P. H. Ketikidis, D. S. Kitz, T. G. Maurides, and B. B. Matschinsky, "Automation of physiologic data presentation and alarms in the post anesthesia care unit," in *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC)* (L. C. Kingsland, ed.), (Washington, D.C), IEEE, IEEE Press, 1989.

[Avent and Charlton, 1990] R. K. Avent and J. D. Charlton, "A critical review of trend-detection methodologies for biomedical monitoring systems," *Crit Rev Biomed Eng*, vol. 17, no. 6, pp. 621–659, 1990.

[Balcerak, 1990] K. J. Balcerak, "Use of computers and clinical data systems," *Aspens Advis Nurse Exec*, vol. 5, no. 4, pp. 4–8, 1990.

[Bankman and Tsitlik, 1991] I. N. Bankman and J. E. Tsitlik, eds., *Proceedings of the Fourth Annual IEEE Symposium on Computer-Based Medical Systems (CBMS)*, vol. 4, (Baltimore, MD), IEEE, IEEE Computer Society Press, May 1991.

[Bishop, 1990] C. W. Bishop, "A new format for the medical record," *M.D. Computing*, vol. 8, no. 4, 1990.

[Bloom, 1986] B. Bloom, *Clinical Information Systems*. New York, NY: Springer-Verlag, 1986.

[Blum and Semmel, 1991] B. J. Blum and R. D. Semmel, "Medical informatics, knowledge, and expert systems," in *Proceedings of the Fourth Annual IEEE Symposium on Computer-Based Medical Systems (CBMS)* (I. N. Bankman and J. E. Tsitlik, eds.), (Baltimore, MD), pp. 212–218, IEEE, IEEE Computer Society Press, 1991.

[Bradshaw *et al.*, 1984] K. Bradshaw, R. Gardner, and T. C. et al, "Physician decision-making — evaluation of data used in a computerized icu," *Intl J Clin Monitoring Comp*, vol. 1, pp. 81–91, 1984.

[Bradshaw *et al.*, 1989] K. E. Bradshaw, D. F. Sittig, R. M. Gardner, T. A. Pryor, and M. Budd, "Computer-based data entry for nurses in the icu," *M.D. Computing*, vol. 6, pp. 274–280, September-October 1989.

[Brown and Krishnamurthy, 1990] P. H. Brown and G. T. Krishnamurthy, "Design and operation of a nuclear medecine picture archiving and communication system," *Semin Nucl Med*, vol. 20, no. 3, pp. 205–224, 1990.

[Bubolz *et al.*, 1989] T. A. Bubolz, J. E. Wennberg, D. J. Malenka, E. S. Fisher, and D. F. McLerran, "Preparing patient histories and unit records from linked medicare data," in *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC)* (L. C. Kingsford, ed.), (Washington, D.C.), pp. 407–413, IEEE, IEEE Press, 1989.

[Carnevale, 1987] F. A. Carnevale, "Computer applications in nursing," *Axon*, vol. 8, no. 1, pp. 269–275, 1987.

[Chaney *et al.*, 1989] R. J. Chaney, F. M. Shipman, and G. A. Gory, "Using hypertext to facilitate information sharing in biomedical research groups," in *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC)* (L. C. Kingsford, ed.), (Washington, D.C.), pp. 350–354, IEEE, IEEE Press, 1989.

[Chnadrasekhar and Price, 1989] A. J. Chnadrasekhar and R. N. Price, "Interactive video in medical education," in *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC)* (L. C. Kingsland, ed.), (Washington, D.C.), pp. 350–354, IEEE, IEEE Press, 1989.

[Cohen, 1990] B. Cohen, "Emerging technologies in the field of healthcare: enhancing the interface to the medical professional," *Ann Acad Med Singapore*, vol. 19, no. 5, pp. 627–639, 1990.

[Collet *et al.*, 1990] C. Collet, L. Martini, M. Lovin, E. Masson, N. Fumai, M. Petroni, A. S. Malowany, F. A. Carnevale, R. D. Gottesman, and A. Rousseau, "Real-time trend analysis for an intensive care unit patient data management system," in *Proceedings of the Third Annual IEEE Symposium on Computer-Based Medical Systems (CBMS)*, (Chapel Hill, NC), pp. 337–344, IEEE, IEEE Computer Society Press, June 1990.

[Dasta, 1990] J. Dasta, "Computers in critical care: opportunities and challenges," *DICP*, vol. 24, no. 11, pp. 1084–1092, 1990.

[Date, 1990] C. J. Date, *An Introduction to Database Systems*. Reading, MA.: Addison-Wesley Publishing Company, fifth ed., 1990.

[Day, 1963] H. Day, "An intensive coronary care area," *Dis Chest*, vol. 44, pp. 423–425, 1963.

[de Dombal, 1986] F. de Dombal, "How do surgeons assimilate information?," *Theor Surg*, vol. 1, pp. 47–54, 1986.

[Denger *et al.*, 1988] S. Denger, D. Cole, and H. Walker, "Implementing an integrated clinical information system," *J Nurs Adm*, vol. 18, pp. 28–34, December 1988.

[Diethrich, 1988] E. B. Diethrich, "Voice recognition and voice synthesis in the cardiac intensive care unit," *Speech Technology*, vol. 4, pp. 46–50, September-October 1988.

[Duhamel *et al.*, 1989] A. Duhamel, P. Roussel, C. Robert, and L. Moussu, "Clinical datafiles and expert systems: Knowledge evaluation from data analysis," in *Proceedings of the 11th Annual International Conference of the IEEE Engineering in Medecine and Biology Society (EMBS)*, pp. 1997–1998, IEEE, IEEE Press, 1989.

[Durbridge *et al.*, 1976] T. Durbridge, F. Edwards, R. Edwards, and M. Atkinson, "Evaluation of benefits of screening tests done immediately on admission to hospital," *Clin Chem*, vol. 22, p. 968, 1976.

[Edmonds, 1990] E. Edmonds, "Human-computer interface evaluation: not user-friendliness but design for operation," *Med Inf (Lond)*, vol. 15, no. 3, pp. 253–260, 1990.

[Ellis *et al.*, 1990] B. W. Ellis, R. C. Rivett, and H. A. Dudley, "Extending the use of clinical audit data: a resource planning model," *BMJ*, vol. 301, no. 6744, pp. 159–162, 1990.

[EMT, 1988] EMTEK, Tempe, AZ, *The EMTEK System 2000: Cost Savings and Benefit Realization*, 1988.

[Ferguson *et al.*, 1989] J. C. Ferguson, C. J. Martin, C. Rayner, and J. R. Mallard, "A computer-based system for monitoring heat and fluid balance in severely burned patients," in *Proceedings of the 11th Annual International Conference of the IEEE Engineering in Medecine and Biology Society (EMBS)*, (Piscataway, NJ), pp. 1982–1983, IEEE, IEEE Press, November 1989.

[Fieldman and Stevens, 1990] C. Fieldman and D. Stevens, "Pilot study on the feasibility of a computerized speech recognition charting system," *Community Dent Oral Epidemiol*, vol. 18, no. 4, pp. 213–215, 1990.

[Fieschi, 1990] M. Fieschi, "Towards validation of expert systems as medical decision aids," *Int J Biomed Comput*, vol. 26, pp. 93–108, July 1990.

[Figler and Stead, 1990] A. A. Figler and S. W. Stead, "The medical information bus," *Biomed Instrum Technol*, vol. 24, pp. 101–111, March-April 1990.

[Fries, 1972] J. Fries, "Time-oriented patient records and a computer databank," *JAMA*, vol. 222, p. 1536, 1972.

[Fries, 1974] J. Fries, "Alternatives in medical record formats," *Med Care*, vol. 12, pp. 871–881, 1974.

[Fumai *et al.*, 1987] N. Fumai, C. Collet, M. Petroni, S. Malowany, A. S. Malowany, F. A. Carnevale, R. D. Gottesman, and A. Rousseau, "A simulator for an intensive care unit patient data management system," in *Proceedings of the 1989 Summer Computer Simulation Conference* (J. K. Clema, ed.), (Austin, Texas), pp. 427–432, The Society for Computer Simulation, The Society for Computer Simulation, July 1987.

[Fumai *et al.*, 1990] N. Fumai, C. Collet, M. Petroni, K. Roger, A. S. Malowany, F. A. Carnevale, R. D. Gottesman, and A. Rousseau, "Evaluation of a database for an intensive care unit patient data management system," in *Proceedings of the Canadian Conference on Electrical and Computer Engineering* (T. A. Gulliver, ed.), (Ottawa, Quebec), pp. 3.4.1–3.4.4, Canadian Society For Electrical and Computer Engineering, Canadian Society For Electrical and Computer Engineering, September 1990.

[Fumai *et al.*, 1991] N. Fumai, C. Collet, M. Petroni, K. Roger, A. Lam, E. Saab, A. S. Malowany, F. A. Carnevale, and R. D. Gottesman, "Database design of an intensive care unit patient data management system," in *Proceedings of the Fourth Annual IEEE Symposium on Computer-Based Medical Systems (CBMS)*, (Baltimore, Maryland), IEEE, IEEE Computer Society Press, May 1991.

[Gardner *et al.*, 1989a] R. M. Gardner, D. F. Sittig, and M. C. Budd, "Computers in the intensive care unit: Match or mismatch?," in *Textbook of Critical Care* (W. C. Shoemaker, S. Ayres, A. Grenvik, P. R. Holbrook, and W. L. Thompson, eds.), ch. 26, Philadelphia: W. B. Saunders Company, second ed., 1989.

[Gardner *et al.*, 1989b] R. Gardner, K. Bradshaw, and K. Hollongsworth, "Computerizing the intensive care unit: current status and future opportunities," *Journal of Cardiovascular Nurse*, vol. 4, pp. 69–78, 1989.

[Gardner, 1986a] R. Gardner, "Artificial intelligence in medecine – is it ready?," *Intl J Clin Monit Comput*, vol. 2, p. 133, 1986.

[Gardner, 1986b] R. Gardner, "Computerized management of intensive care patients," *MD Computing*, vol. 3, no. 1, p. 36, 1986.

[Gibbs, 1989] R. Gibbs, "The present and future medicolegal importance of record keeping in anesthesia and intensive care: the case for automation," *J Clin Monit*, vol. 5, no. 4, pp. 251–255, 1989.

[Goldberg *et al.*, 1989] M. Goldberg, J. Robertson, G. Belanger, N. Georganas, J. Mastronardi, S. Cohn-Sfetcu, R. Dillon, and J. Tombaugh, "A multimedia medical communication link between a radiology department and an emergency department," *J Digit Imaging*, vol. 2, pp. 92–98, May 1989.

[Gro, 1991] Groupe LGS Inc, *Comptes Rendu du Salon Annuel Informatique Santé*, 1991.

[Halford *et al.*, 1989] G. Halford, M. Burkes, and T. A. Pryor, "Measuring the impact of bedside terminals," *Nursing Management*, vol. 20, no. 7, pp. 41–45, 1989.

[Hammond and Stead, 1988] W. E. Hammond and W. W. Stead, "Bedside terminals: An overview," *M.D. Computing*, vol. 5, no. 1, pp. 5–6, 1988.

[Hammond *et al.*, 1991a] J. Hammond, H. Johnson, C. Ward, R. Varas, R. Dembicki, and E. Marcial, "Clinical evaluation of a computer-based patient monitoring and data management system," *Heart Lung*, vol. 20, no. 2, pp. 119–124, 1991.

[Hammond *et al.*, 1991b] J. Hammond, H. M. Johnson, R. Vanes, and C. G. Ward, "A qualitative comparison of paper flowsheets vs a computer-based clinical information system," *Chest*, vol. 99, pp. 155–157, January 1991.

[Hasman *et al.*, 1988] A. Hasman, R. M. J. Silkens, P. Zinken, A. B. M. F. Karim, and R. F. Westerman, "Adamo revisited: an interpretive review of a data management system," *International Journal of Biomedical Computing*, vol. 23, pp. 21–32, 1988.

[Haug *et al.*, 1990] P. J. Haug, D. L. Ranum, and P. R. Frederick, "Computerized extraction of coded findings from free-text radiologic reports," *Radiology*, vol. 174, no. 2, pp. 543–548, 1990.

[Hea, 1991] Health Data Sciences Corporation, *Comptes Rendu du Salon Annuel Informatique Santé*, 1991.

[Hendrickson *et al.*, 1991] G. Hendrickson, J. B. Kelly, and L. Citrin, "Computers in oncology nursing: Present use and future potential," *Oncology Nursing Forum*, vol. 18, pp. 715–723, May-June 1991.

[Hew, 1986] Hewlett Packard, Manual Part No. 78580-91809-8, *Hewlett Packard HP 78580 CarePort Programmable SDN Interface*, 1986.

[Hew, 1988] Hewlett Packard, Manual Part No. 78580-42308-6, *Hewlett Packard PDMS/CareNet System: Programmers Reference Guide*, 1988.

[Hilberman, 1975] M. Hilberman, "The evolution of the intensive care unit," *Critical Care Medecine*, vol. 3, pp. 159–163, 1975.

[Holtermann *et al.*, 1990] W. Holtermann, M. Knoch, H. Pfeiffer, E. E. Muller, and H. Lennartz, "Marburger concept for computer-aided acquisition, processing and documentation of patient data in the intensive care unit," *Int J Clin Monit Comput*, vol. 7, no. 1, pp. 7–13, 1990.

[Hop, 1991] Hopital Notre-Dame, *Comptes Rendu du Salon Annuel Informatique Santé*, 1991.

[Humphreys and Lindberg, 1989] B. L. Humphreys and D. A. B. Lindberg, "Building the unified medical language system," in *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC)* (L. C. Kingsford, ed.), (Washington, D.C.), pp. 475–482, IEEE, IEEE Press, 1989.

[IBM, 1990] IBM, Manual No. 0170292, S01F-0292-00, *IBM Operating System/2 Extended Edition Version 1.3 Database Manager Programming Guide and Reference*, 1990.

[Ikehira *et al.*, 1990] A. Ikehira, T. Matsumoto, T. Iinuma, T. Yamasaki, K. Fukuhisa, H. Tsunemoto, F. Shishido, Y. Kubo, K. Inamura, and Y. Tateno, "Analysis of bone scintigram data using speech recognition reporting system — data analysis with speech recognition systems," *Radiat Med*, vol. 8, no. 1, pp. 8–12, 1990.

[Jajodia and Sandhu, 1991] S. Jajodia and R. Sandhu, "Toward a multilevel secure relational data model," in *Proceedings of the 1991 ACM SIGMOD International Conference on Management of Data* (J. Clifford and R. King, eds.), (Denver, Colorado), pp. 50–59, ACM SIGMOD, ACM, 1991.

[James *et al.*, 1990] J. C. James, N. S. Gatenberg, and G. R. Hageman, "A sample computer system for physiological data acquisition and analysis," *Comput Biol Med*, vol. 20, no. 6, pp. 407–413, 1990.

[Joslyn, 1989] J. S. Joslyn, "A computerized hospital to physician office link based on readily available technology," in *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC)* (L. C. Kingsford, ed.), (Washington, D.C.), pp. 466–474, IEEE, IEEE Press, 1989.

[Kari *et al.*, 1990] A. Kari, E. Ruokonen, and J. Takala, "Comparison of acceptance and performance of automated and manual data management systems in intensive care," *Int J Clin Monit Comput*, vol. 7, no. 3, pp. 157–162, 1990.

[King and Smith, 1990] P. H. King and B. E. Smith, "An ibm at based monitoring system with touchscreen input," *Int J Clin Monit Comput*, vol. 7, no. 2, pp. 107–11, 1990.

[Kirschner, 1930] M. Kirschner, "Zum neubau der chirurgischen universitaetsklinik tuebingen," *Der Chirurg*, vol. 2, pp. 54–XX, 1930.

[Knaus and et al, 1985] W. Knaus and E. D. et al, "Apache ii: A severity of disease classification system," *Crit Care Medecine*, vol. 13, pp. 818–829, 1985.

[Knight, 1990] J. C. Knight, "Issues of software reliability in medical systems," in *Proceedings of the Third Annual IEEE Symposium on Computer-Based Medical Systems (CBMS)* (June, ed.), (Chapel Hill, NC), pp. 153–160, IEEE, IEEE Computer Society Press, 1990.

[Korpman, 1990] R. A. Korpman, "Patient care automation: the future is now. part 3. the five rules of automation," *Nurs Econ*, vol. 8, pp. 345–349, September-October 1990.

[Kriewall and Long, 1991] T. J. Kriewall and J. M. Long, "Computer-based medical systems," *IEEE Computer*, vol. 24, pp. 9–12, March 1991.

[Lal-Gabe, 1990] A. Lal-Gabe, "Hazards analysis and its application to build confidence in software test results," in *Proceedings of the Third Annual IEEE Symposium on Computer-Based Medical Systems (CBMS)* (June, ed.), (Chapel Hill, NC), pp. 129–136, IEEE, IEEE Computer Society Press, 1990.

[Large, 1990] W. Large, "Calculating haemodynamic parameters and interpreting arterial blood gas samples using a pocket computer," *Intensive Care Nurse*, vol. 6, no. 4, pp. 196–199, 1990.

[Ledley and Lusted, 1959] R. Ledley and L. Lusted, "Reasoning foundations of medical diagnosis," *Science*, vol. 130, pp. 9–21, 1959.

[Lee and Lee, 1991] E. T. Lee and J. Y. Lee, "Medical databases," in *Proceedings of the 13th Annual International Conference of the IEEE Engineering in Medecine and Biology Society (EMBS)* (J. H. Nagel and W. M. Smith, eds.), (Orlando, FL), pp. 1369–1370, IEEE, IEEE Press, November 1991.

[Les, 1991] Les Ordinateurs Hypocrat Inc, *Comptes Rendu du Salon Annuel Informatique Santé*, 1991.

[Leyerle *et al.*, 1990] B. J. Leyerle, M. LoBue, and M. Shabot, "Integrated computerized databases for medical data management beyond the bedside," *Int J Clin Monit Comput*, vol. 7, no. 2, pp. 83–9, 1990.

[Lusted, 1968] L. Lusted, *Introduction to Medical Decision Making*. Springfield, Illinois: Thomas, 1968.

[Lyons and Gumpert, 1990] C. Lyons and R. Gumpert, "Medical audit data: counting is not enough," *BMJ*, vol. 300, no. 6739, pp. 1563–6, 1990.

[Malowany *et al.*, 1989] S. Malowany, C. Collet, N. Fumai, M. Petroni, A. S. Malowany, F. A. Carnevale, R. D. Gottesman, and A. Rousseau, "Data base aspects of a patient data management system for an icu," in *Proceedings of the Canadian Conference on Electrical and Computer Engineering* (V. K. Bhargava, ed.), (Montreal, Quebec), pp. 586–589, Canadian Society For Electrical and Computer Engineering, Canadian Society For Electrical and Computer Engineering, September 1989.

[Martin, 1989] G. L. Martin, "The utility of speech input in user-computer interfaces," *International Journal of Man-Machine Studies*, vol. 30, pp. 355–375, 1989.

[McDonald and Tierney, 1988] C. McDonald and W. Tierney, "Computer-stored medical records," *JAMA*, vol. 259, pp. 3433–3440, 1988.

[Meghini *et al.*, 1991] C. Meghini, F. Rabitti, and C. Thanos, "Conceptual modeling of multimedia documents," *IEEE Computer*, vol. 24, pp. 23–30, October 1991.

[Miller and Giuse, 1991] R. A. Miller and N. B. Giuse, "Medical knowledge bases," *Acad Med*, vol. 66, no. 1, pp. 15–17, 1991.

[Murchie and Kenny, 1988] C. J. Murchie and G. N. C. Kenny, "A comparison of keyboard, light pen and voice recognition as methods of data input," *International Journal of Clinal Monitoring and Computing*, vol. 5, no. 4, pp. 243–246, 1988.

[Myers, 1991] B. A. Myers, "Demonstrational interfaces: A step beyond direct manipulation," in *People and Computers VI* (D. Diaper and N. Hammond, eds.), pp. 11–30, Cambridge, England: Cambridge University Press, 1991.

[Nagel and Smith, 1991] J. H. Nagel and W. M. Smith, eds., *Proceedings of the 13th Annual International Conference of the IEEE Engineering in Medecine and Biology Society (EMBS)*, vol. 13, (Orlando, FL), IEEE, IEEE Press, November 1991.

[Nightingale, 1863] F. Nightingale, *Notes on a Hospital*. London: Longman, third ed., 1863.

[Nolan-Avila and Shabot, 1987] L. Nolan-Avila and M. Shabot, "Life without computers in the icu," *Critical Care Nurse*, vol. 7, pp. 80–83, May-June 1987.

[O'Neil and Glowinski, 1990] M. O'Neil and A. Glowinski, "Evaluating and validating very large knowledge-based systems," *Med Inf (Lond)*, vol. 15, pp. 237–251, July-September 1990.

[Osler *et al.*, 1990] T. M. Osler, G. Reddy, D. Fletcher, G. B. Demarest, F. W. Clevenger, J. M. Nachbar, D. J. Downey, and D. E. Fry, "A computerized approach to injury description," *J Trauma*, vol. 30, pp. 983–987, August 1990.

[Ozdamar *et al.*, 1989] O. Ozdamar, I. Yaylali, P. Jayakar, and C. N. Lopez, "Multilevel neural network system for eg spike detection," in *Proceedings of the Fourth Annual IEEE Symposium on Computer-Based Medical Systems (CBMS)* (I. N. Bankman and J. E. Tsitlik, eds.), (Baltimore, MD), pp. 272–279, IEEE, IEEE Computer Society Press, May 1989.

[Padjen and Salasidis, 1988] A. Padjen and R. Salasidis, "The pdb system," in *Proceedings of the 12th Annual Symposium on Computer Applications in Medical Care (SCAMC)*, pp. 31–33, 1988.

[Paganelli, 1989] B. E. Paganelli, "Criteria for the selection of a bedside information system for acute care units," *Computers in Nursing*, vol. 7, pp. 214–221, September-October 1989.

[Panisset *et al.*, 1989] J.-F. Panisset, D. Lambidonis, N. Khoury, S. Malowany, A. S. Malowany, F. A. Carnevale, R. D. Gottesman, and A. Rousseau, "An intensive care unit patient data management system," in *Proceedings of the Annual Graphics Interface Conference* (M. Wein and E. M. Kidd, eds.), (London ON), pp. 275–282, Canadian Information Processing Society, Canadian Information Processing Spciety, June 1989.

[Parkin *et al.*, 1990] A. Parkin, H. Norwood, A. Erdentug, and A. J. Hall, "Optical disk archiving using a personal computer: a solution to image storage problems in diagnostic imaging departments," *J Med Eng Technol*, vol. 14, no. 2, pp. 55–59, 1990.

[Perry, 1990] C. A. Perry, "Knowledge bases in medecine: a review," *Bull Med Libr Assoc*, vol. 78, pp. 271–282, July 1990.

[Pesce, 1988] J. Pesce, "Bedside terminal: Medtake," *M.D. Computing*, vol. 5, pp. 16–21, January-February 1988.

[Petroni *et al.*, 1991] M. Petroni, C. Collet, N. Fumai, K. Roger, A. Lam, A. S. Malowany, F. A. Carnevale, and R. D. Gottesman, "A speech interface for bedside data entry in an intensive care unit," in *Proceedings of the 13th Annual International Conference of the IEEE Engineering in Medecine and Biology Society (EMBS)* (J. H. Nagel and W. M. Smith, eds.), vol. 13, (Orlando, FL), pp. 1373–1374, IEEE, IEEE Press, November 1991.

[Pollacia, 1989] L. F. Pollacia, "A survey of discrete event simulation and state of the art discrete event languages," *ACM Simulation Digest*, vol. 31, no. 3, pp. 8–25, 1989.

[Pollack *et al.*, 1987] M. Pollack, U. Ruttimann, and P. Getson, "Accurate prediction of the outcome of pediatric intensive care: A new quantitative method," *New England Journal of Medecine*, vol. 317, pp. 134–138, 1987.

[Prakash *et al.*, 1991] O. Prakash, N. Govindarajan, S. Meiyappan, and K. S. Sundar, "Workstations for the operating room and critical care," in *Proceedings of the 13th Annual International Conference of the IEEE Engineering in Medecine and Biology Society (EMBS)* (J. H. Nagel and W. M. Smith, eds.), (Orlando, FL), pp. 1226–1227, IEEE, IEEE Press, November 1991.

[Price and Chandrasekhar, 1989] R. N. Price and A. J. Chandrasekhar, "Department of medecine local area network: A strategic solution for the 1990's," in *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC)* (L. C. Kingsford, ed.), (Washington, D.C.), pp. 462–466, IEEE, IEEE Press, 1989.

[Rathe *et al.*, 1989] R. Rathe, L. Cope, R. A. Greenes, and J. Glaser, "System architecture for a clinical workstation providing point of use knowledge access," in *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care*

(SCAMC) (L. C. Kingsford, ed.), (Washington, D.C.), pp. 669–674, IEEE, IEEE Press, 1989.

[Read *et al.*, 1990] C. Read, E. H. Buder, and R. D. Kent, "Speech analysis systems: a survey," *J Speech Hear Res*, vol. 33, pp. 363–374, June 1990.

[Roger *et al.*, 1991] K. Roger, C. Yien, M. Petroni, C. Collet, N. Fumai, A. S. Malowany, F. A. Carnevale, and R. D. Gottesman, "A nursing workload manager for a patient data management system," in *Proceedings of the 13th Annual International Conference of the IEEE Engineering in Medecine and Biology Society (EMBS)* (J. H. Nagel and W. M. Smith, eds.), vol. 13, (Orlando, FL), pp. 1371–1372, IEEE, IEEE Press, 1991.

[Russler, 1991] D. C. Russler, "Integration of distributed medical information sources on the clinical workstation," in *Proceedings of the 13th Annual International Conference of the IEEE Engineering in Medecine and Biology Society (EMBS)* (J. H. Nagel and W. M. Smith, eds.), vol. 13, (Orlando, FL), pp. 1216–1217, IEEE, IEEE Press, 1991.

[Ruth *et al.*, 1947] H. Ruth, F. Hauvgen, and D. Grave, "Anesthesia study commission: Findings of eleven years' activity," *JAMA*, vol. 35, pp. 881–XXX, 1947.

[Saab *et al.*, 1992] E. Saab, N. Fumai, M. Petroni, K. Roger, C. Collet, A. S. Malowany, F. A. Carnevale, and R. D. Gottesman, "Data modeling and design of a patient data management system in an intensive care unit," in *Proceedings of the Fifth Annual IEEE Symposium on Computer-Based Medical Systems (CBMS)*, (Durham, North Carolina), IEEE, IEEE Computer Society Press, 1992.

[Safran *et al.*, 1990] C. Safran, D. Porter, C. D. Rury, F. R. Herrmann, J. Lightfoot, L. H. Underhill, H. L. Bleich, and W. V. Slack, "Clinquery: searching a large clinical database," *M.D. Computing*, vol. 7, no. 3, pp. 144–153, 1990.

[Savona-Ventura *et al.*, 1990] C. Savona-Ventura, E. S. Grech, A. Grech, and P. Fenech-Gonzi, "Computerization of data: a person-based medical record system," *Int J Gynaecol Obstet*, vol. 32, no. 3, pp. 247–254, 1990.

[Scherrer *et al.*, 1990] J. R. Scherrer, R. H. Baud, and D. Hochstrasser, "An integrated hospital information system in geneva," *M.D. Comput*, vol. 7, no. 2, pp. 81–89, 1990.

[Scherrer, 1990] J. R. Scherrer, "New architectures destined for hospital computer networks opening the medical world to more communication facilities of every kind," *Schweiz Med Wochenschr*, vol. 120, no. 49, pp. 1866–1871, 1990.

[Schmidt *et al.*, 1989] R. Schmidt, A. Schurholz, and M. Ruger, "Create! - simulation aided development of control software for automated material flow systems," in *Proceedings of the 1989 Summer Computer Simulation Conference* (J. K. Clema, ed.), (Austin, Texas), pp. 64–69, The Society for Computer Simulation, The Society for Computer Simulation, July 1989.

[Schwid *et al.*, 1990] H. Schwid, C. T. Olson, P. Wright, and P. P. Freund, "Microcomputer-based data acquisition system for clinical research," *J Clin Monit*, vol. 6, no. 2, pp. 141–146, 1990.

[Selby, 1990] T. Selby, "Nursing care goes high tech," *Am-Nurse*, vol. 22, no. 7, pp. 1–21, 1990.

[Shetler, 1990] T. Shetler, "Birth of the blob," *BYTE*, vol. 15, pp. 221–226, February 1990.

[Shneiderman, 1987] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley Publishing Company, 1987.

[SID, 1991] SIDOCI, *Comptes Rendu du Salon Annuel Informatique Santé*, 1991.

[Sinclair and Ross, 1990] D. G. Sinclair and D. W. Ross, "Recombinant dna laboratory data manager using dbase iii+," *Biotechniques*, vol. 8, pp. 666–669, June 1990.

[Sivak *et al.*, 1989] E. D. Sivak, J. S. Gochberg, and D. M. Frank, "Lessons to be learned from the continuing epic of computerizing the intensive care unit," in *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC)* (L. C. Kingsland, ed.), (Washington, D.C.), pp. 605–610, IEEE, IEEE Press, 1989.

[Soligen and Shabot, 1988] S. Soligen and M. M. Shabot, "A 32 key keyboard for the hp pdms," *Int J Clin Monit Comput*, vol. 5, no. 1, 1988.

[Southerton, 1989] A. Southerton, *Advanced OS/2 Presentation Manager Programming: The Graphics Programming Interface*. Reading, MA: Addison-Wesley Publishing Company, 1989.

[Stead *et al.*, 1983] W. Stead, W. Ilammond, and M. Straube, "A chartless record – is it adequate?," *J Med Syst*, vol. 7, p. 103, 1983.

[Stoeckler and Ellis, 1989] J. S. Stoeckler and L. B. M. Ellis, "A laptop computer application for neonatal intensive care," in *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC)* (L. C. Kingsford, ed.), (Washington, D.C.), pp. 569–574, IEEE, IEEE Press, 1989.

[Sys, 1991] Systemes Purkinje Inc, *Comptes Rendu du Salon Annuel Informatique Santé*, 1991.

[Tachakra *et al.*, 1990] S. Tachakra, D. Potts, and A. Idowu, "Evaluation of a computerized system for medical records in an accident and emergency department," *Int J Clin Monit Comput*, vol. 7, no. 3, pp. 187–191, 1990.

[Tullis, 1981] T. S. Tullis, "An evaluation of alphanumeric, graphic and color information displays," *Human Factors*, pp. 541–550, 1981.

[Varde *et al.*, 1991] A. S. Varde, K. L. Massey, and H. C. Wood, "A clinical laboratory expert system for the diagnosis of thyroid dysfunction," in *Proceedings of the 13th Annual International Conference of the IEEE Engineering in Medecine and Biology Society (EMBS)* (J. H. Nagel and W. M. Smith, eds.), (Orlando, FL), pp. 1288–1289, IEEE, IEEE Press, November 1991.

[Venditto, 1991] G. Venditto, "Relational databases: Easy access, programming power," *PC Magazine*, May 28 1991.

[Weil *et al.*, 1976] M. Weil, H. Shubin, and R. Carlson, *The new practice of critical care medecine*. New York: Harper and Row, 1976.

[Weil *et al.*, 1989] M. H. Weil, M. V. Planta, and F. C. Rackow, "Critical care medecine: Introduction and historical perspective," in *Textbook of Critical Care* (W. C. Shoemaker, S. Ayres, A. Grenvik, P. R. Holbrook, and W. L. Thompson, eds.), ch. 1, pp. 1–5, Philadelphia, PA: W. B. Saunders Company, second ed., 1989.

[Whiting-O'Keefe *et al.*, 1980] Q. Whiting-O'Keefe, D. Simborg, and W. Epstein, "A controlled experiment to evaluate the use of time-oriented summary medical records," *Med Care*, vol. 8, pp. 842–852, 1980.

[Whiting-O'Keefe *et al.*, 1985] Q. Whiting-O'Keefe, D. Simborg, and W. E. et al, "A computerized summary medical record system can provide more information than the standard medical record," *JAMA*, vol. 254, p. 1185, 1985.