National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

A Rule-Based System for Vital Sign Monitoring in Intensive Care

Abolfazl Hosseinzadeh

B. Eng., (Tehran University), 1987

Department of Electrical Engineering

McGill University

Montréal

November, 1993

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 0-315-94273-8

Canada

I dedicate this work to my parents

for showing me that knowledge is one of life's hidden jewels.

Abstract

The implementation of a knowledge-based expert system to check on inpatients' conditions is described. In our development of a bedside computerized ICU, graphical user interface (GUI) and graphical visualization have been extensively exploited to facilitate the natural and effective representation of clinical information for patient care, the transition from the manual system to computerized system and users interaction with the system. The input comprises the heart rates, the blood pressures, and the central venous pressures of several inpatients.

The algorithm is made up of two parts. The first one, written in C acquires and classifies the data. The second part compares the classified data with the medical information stored in its data base, and draws a conclusion regarding the patient's condition. C Language Integrated Production System (CLIPS) 5.0 is the knowledge-based expert system utilized. The results obtained from the Expert system are displayed using the Simple Raster Graphics Package (SRGP). This system has been implemented and tested on a variety of inputs, and the results are depicted.

Acknowledgements

I would like to express my sincere gratitude to Prof. Alfred Malowany. I am indebted a great deal to him, for his countless hours of interesting conversation and guidance. Without his encouragement and invaluable input, the completion of this thesis would not have been possible.

I truly appreciate Prof. F. Hassani for his guidance and assistance in continuing my education.

I am grateful to Prof. P. Belanger, Prof. D. Lowther, Prof. R. De Mori, Prof M. Levine, and Prof. I. Leszkowicz for helping me understand the underlying theories of Electrical Engineering.

I would like to thank my colleagues and the staff of McRCIM and the Department of Electrical Engineering of McGill for their support.

I also wish to express my appreciation to my friends Mr. Mehron Vaezi, Mr. Reza Fardshisheh, Mr. Antonious Haakman, Miss Irene Karantanis, Mr. Alain Cordier, and Mr. Steve Malowany for advising and assisting me in preparing this manuscript.

This work would not have been possible without the support of the Ministry of Culture and Higher Education of Iran. Special thanks to Mr. Khatai, Mr. Naghash, Mr. Zolfaghari, and Mr. Akhavani for their guidance.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

The past century has been revolutionary in scientific and technological achievements. Paradigms of scientific discovery are now a commonplace. In the past several decades, this revolution has stirred renewed debate on not only the essence of science and its methods, but also on the nature of cognition and belief. One of the most enduring of the philosophical dichotomies on such issues, extending back to the time of Confucius, concerns whether truth is best comprehended in terms of one big thing or in term of many little things.

This work addresses one of the most challenging and difficult areas of contemporary decision; that of the world of clinical problem solving, though these decisions are supported by an expanding but always tentative body of scientific knowledge.

This thesis presents an implementation written in C and in CLIPS 5.0 language and an evaluation of their execution results. The system aims at monitoring the condition of inpatients, and subsequently, it infers an estimate of their medical conditions. The algorithm was designed on the forward chaining method of inferencing. This can be defined as a problem-solving search strategy that starts with features of the acquired data and infers their immediate consequences. Those consequences are added to the available data and further inferences are drawn. This continues until the goals or targets of the problem have been reached. Other terms for this are data-directed search and antecedent reasoning. The name is based on using the left-hand-side (LHS) of logical rules as a testing point, and if satisfied going forward to carry out the actions specified in the right-hand-side (RHS), and so on [Rychener, 1988]. Then the results are visually displayed. In the following sections we shall briefly introduce the expert systems and computer-based medical

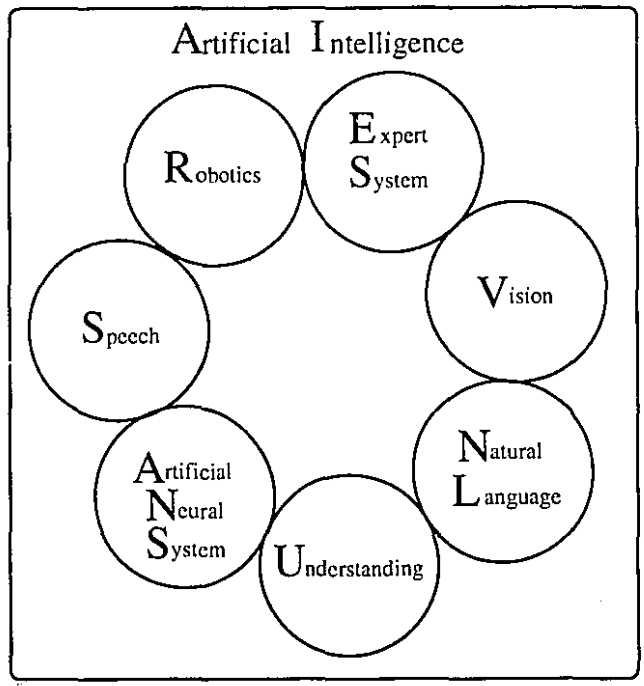

Figure 1.1: Some areas of Artificial Intelligence [Giarratano, 1989].

systems.

1.1 Artificial Intelligence and Knowledge Engineering

Artificial Intelligence (AI) is the endeavor to construct computer systems that perform tasks (especially intellectual ones) that are considered to require intelligence. The main areas of pure AI research are problem solving and search, common-sense reasoning and deduction, representating of knowledge, learning, and system architectures for AI. Fig. 1.1 shows many areas of applications of AI [Giarratano, 1989]. Areas such as robotics [Cinquin, 1992, Davies, 1992, Martelli, 1992], natural language processing [McCray, 1991], image understanding (vision) [Cinquin, 1992], and expert systems [Gururajan, 1992, Ravaux, 1992] are considered applications of the core AI concepts [Rychener, 1988].

Since the early 1970's, AI concentrated on general problem solving techniques [Newell, 1972], and this resulted in many studies of the strategies used in stylized

problems and game playing with deterministic rules. In the following two decades, a trend of research of problem solving in real-life domains, particularly in chemistry, medicine, genetics, and geology has emerged.

Several AI methods have been developed over the past decade for representing the structure of knowledge needed to characterize clinical decisions. They include causal networks [Weiss, 1983, Clancey, 1992, Summers, 1992], production rules [Shortliffe, 1976, Goodall, 1992], prototypical templates or frames [Pauket, 1976, Gururajan, 1992], and hierarchical networks [Pople, 1975, Moret-Bonillo, 1992]. Several consultation systems have been built based on the knowledge of clinical experts (or groups of experts) and later tested with substantial numbers of clinical cases. The first major systems were in the areas of infectious diseases -MYCIN [Shortliffe, 1973], and internal medicine -INTERNIST [Pople, 1977], PIP [Szolovitz, 1978], ophthalmology -CASNET [Weiss, 1978]. The last decade studies combined representational and inferential ideas from several systems [Reggia, 1978, Fagan, 1984, Blum, 1986, Catanzaite, 1986]. There have also been efforts to develop generalized schemes for assisting the expert in rapidly encoding the reasoning concepts and decision rules [Alonso-Betanzos, 1991], developed from two of the early AI systems. These are the EXPERT [Weiss, 1979] and EMYCIN [VanMelle, 1978] schemes that evolved from the CASNET and MYCIN programs respectively, the AGE system [Nii, 1979], KIDS [Ovalle, 1991, Derder, 1992], and ECG [Laguna, 1992]. Section 1.2.4 will present an overview of Artificial Intelligence in medicine.

1.1.1 Knowledge-Based Systems

In the past few decades, a new computing technology has emerged from research in Artificial Intelligence for application to a variety of technical domains. This technology is called expert systems or knowledge-based systems. The expert system takes knowledge from human experts and represents it as a knowledge-base, which can then be processed to solve difficult problems in the same way the expert

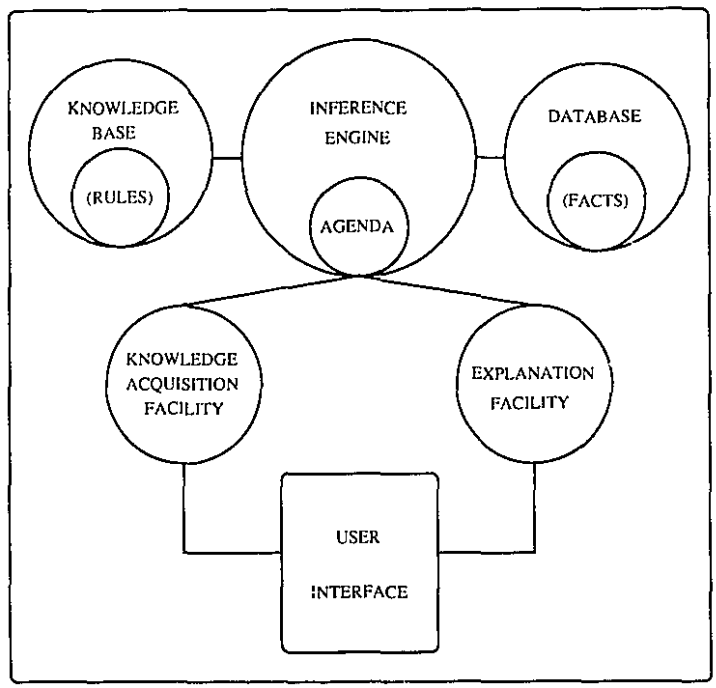

Figure 1.2: Structure of a Rule-Based Expert System.

would. The elements of a typical expert system are shown in Fig. 1.2. An expert system consists of the following components:

1. User interface - the mechanism by which the user and the expert system communicate.

2. Explanation facility - explains the reasoning of the system to a user.

3. Database - a global collection of facts used by the rules.

4. Inference engine - makes inferences by deciding which rules are satisfied by facts, prioritizes the satisfied rules, and executes the rule with the highest priority.

5. Agenda - a prioritized list of rules created by the inference engine, whose patterns are satisfied by facts in working memory.

6. Knowledge acquisition facility - an automatic way for the user to enter knowledge in the system rather than by having the knowledge engineer explicitly

code the knowledge.

Evaluation should be considered to be a major topic of interest during the design and development of a knowledge based system. A number of a evaluation procedures may be distinguished: verification, validation, laboratory evaluation, and field evaluation [Daalen, 1992].

1.1.2 Rule-Based Systems

One of the most popular type of expert system today is the rule-based system. Rules are popular for a number of reasons.

- Similarity to the human cognitive process: Rules appear to be a natural way of modeling how humans solve problems [Newell, 1972]. The simple IF THEN representation of rules makes it easy to explain to experts the structure of the knowledge that you are trying to elicit from the rules.
- Modular nature: This makes it easy to encapsulate knowledge and expand the expert system by incremental development.
- Explanation facilities: It is easy to build explanation facilities with rules since the antecedents of a rule specify exactly what is necessary to activate the rule. By keeping track of which rules have fired, an explanation facility can present the chain of reasoning that led to a certain conclusion.

Other advantages of rules are described in [Hayes-Roth, 1987].

In order to accomplish useful work, an expert system must have rules as well as facts. The term *rule* denotes a *condition/action pair* or an IF/THEN *statement*. The condition or IF-part specifies a condition or context in which a certain operation can be performed, and the second or THEN-part specifies the operation itself or

its results. Each rule is identified by a name. Following the name is the IF-part of the rule. The section of the rule between the IF and THEN part of the rule is called by various names such as the *antecedent, conditional part, pattern part,* or *left-hand-side* (LHS). The individual condition is called a *conditional element* or a *pattern.* The following rule, for example, expresses a patient's condition

IF the patient's heart rate, blood pressure, and
central venous pressure are low

THEN the patient's condition is *Hypovolemia, Braddyarrhythmia,* and *Pump Failure*

A very popular form for expressing a rule consists of a *left-hand-side* (LHS), which represents the condition part, and a *right-hand-side* (RHS), which represents the action part [Flemming, 1988], with an arrow used to separate the two sides form each other, giving the entire rule in the form:

$$\text{LHS} \longrightarrow \text{RHS}$$

One way of building an expert system is to use one of the programming languages suitable for AI. *C Language Integrated Production System* (CLIPS) is one of the languages currently used. Chapter 3 reviews CLIPS fundamental concepts.

1.2 Computer-Based Medical Systems

During the last several years significant research effort has been devoted to developing computer systems for processing clinical knowledge. Such systems are referred to as *knowledge-based* programs because they are built upon a database of medical knowledge [Gilbert, 1990]. The knowledge they contain is usually *synthesized* in the sense that it has been prepared by one or more medical experts from multiple existing information sources such as text-books, the literature, or

personal experience.

In medical applications, the basic idea behind these programs is to capture diagnostic and therapeutic decision-making knowledge in a computer program [Jordan, 1991, Krusinska, 1991]. The expectation is that such automated expertise, together with the computer's powerful logic and memory resources, can then be applied to many useful tasks. To be successful, these systems must be coordinated with [Stanton, 1993] proper accompanying administrative legal and labour structures.

The predominant approach to creating knowledge-based systems has been focused primarily on computer use of knowledge to assist in medical decision making [Reggia, 1985, Nykanen, 1992]. While those developing such computer-assisted medical decision-making (CMD) systems [Montgomery, 1991, Lazarro, 1992] have paid some attention to providing explanatory and educational abilities [Abel, 1992], most of the effort in this work has concentrated on the decision-making process itself [Grygotis, 1991, Hadzikadic, 1991].

1.2.1 Historical Overview

The application of computers to medicine began as the first computer became commercially available. Naturally, the history of medical computing has been closely tied to the growing capability of computers. In general, there has been a lag of five years between a technical advance in the computer field and its use in medical computing [Blum, 1987].

The computer technology of the 1950's saw the transition from vacuum tubes to transistors. Computers were large, required considerable support in both maintenance and environmental control, and were generally limited to running one job at a time. Operating systems were developed to facilitate the use of the computer by programmers and operators. Tools to write programs grew from assembly

languages to high-order languages such as COBOL and FORTRAN.

Two major parallel developments were seen in the 1960's: the use of integrated circuits and the growth of computer science. The result was the manufacture of large, reliable computers. Consequently, multiprogramming operating systems were developed and smaller computers (minicomputers) were also manufactured.

The 1970's saw an unanticipated revolution in very large scale integration (VLSI). Computers of great reliability, small size and low cost were produced and previously impractical applications became cost-effective.

The 1980's brought some new characteristics. First, computers were no longer confined to the computer room; they were integral components of watches, cameras, and kitchen appliances. Packaged as microcomputers or personal computers, it became possible for the hobbyist to own the equivalent computer power of the largest systems of the 1950's. Because of the lowered cost and smaller packaging, computer architectures were also undergoing change. Networks allowed the integration of multiple systems.

Dreams of the 1990's could include an invisible butler, or an army of quick servants waiting to put your house in order! Within the next five years, we may find ourselves surrounded by *thinking* devices that may help us screen our mail and telephone calls. They will monitor newspapers and magazines and provide a digest of material that fits our interests and handle our appointments with the skill of a good secretary [Chandler, 1992].

1.2.2 Hospital Information Systems

A hospital is a natural environment for an automated system. It is relatively large, labor intensive, information oriented, and dependent upon communications among different operational units. Difficulties associated with the implementation

of an automated system include the cost and reliability of computers, the lack of standards among hospitals, and the personal nature of health care delivery [D'Atri, 1992].

Hospital Information Systems (HISs) were first developed to process administrative data but very soon there was a strong request to include the management of clinical data for true clinical purposes [Ilomaki, 1992]. The technology used in the first generation of HISs, however, was not yet sufficiently well developed to make a complete system feasible. At the moment the clinicians are close to having access, in an acceptable time and in a reasonable format, to an integrated representation and management of all the medical data of a given patient [Catarci, 1990, Catarci, 1992].

The recent information and communication technologies enable the development of a new generation of HISs that overcome the limitations encountered until now: lack of a complete integration of heterogenous data, lack of advanced user interface, limited computer performances. Considering the structure of the hospital and the need for evolution and change, the design of *fully integrated* HISs had to be postponed, settling for the achievement of information systems realizing integration mainly at departmental level. Such Medical Information Systems (MISs) have as goals the development of integrated departmental systems allowing their users to manipulate and efficiently process several different kinds of medical data in a user-friendly and homogeneous way [Celler, 1990, Catarci, 1992].

1.2.3 Data Base Management Systems

Modern medical information systems have at their core a data base management system [Orhanoudakis, 1991] which permits flexible addition and deletion of data items. The most sophisticated of these data base management systems incorporate a dictionary driver concept in which the data base itself asks a series of questions

about the data which, once answered, establish the *dictionary*. The dictionary *drives* all other programs i.e. enter, edit, search, report, statistics, etc. whenever a change is made in the dictionary, it drives required changes in all other programs.

Most data base management systems have been developed to deal with single encounters [Entine, 1984]. Only a few have been specifically designed to deal with data which changes over time, as patients change [Faulkner, 1988]. These time oriented data base management systems permit clinicians, administrators, researchers, and educators to track patients and their progress, follow a course of treatment, keep a service history, analyze staff or trainee activities, keep inventories, etc. [Greist, 1987, Hudson, 1992].

Design and development of any software project may be divided up into several steps [Pressman, 1987]. Data base life cycle design consists of five basic steps [Hughes, 1988]:

- requirement analysis
- data modeling
- implementation
- testing
- maintenance

The quality of a data base system is significantly influenced by the quality of the data model, and thus data modeling occupies an important position in the data base life cycle [Alnahi, 1992].

1.2.4 Artificial Intelligence in Medicine

In the early 1970's, there emerged considerable discouragement and disappointment with respect to computer applications in clinical medicine. At about the

same time, similar discouragement occurred among the supporters of the Artificial Intelligence community - language translation had not proceeded well, and image processing was far more complex than it had seemed earlier. Though perhaps only coincidental, it appears that at about this time the two communities recognized a core of mutual interests. The merging of these interests to date has focused primarily on expert-level consultation programs, but it has also been concerned with how physicians make clinical decisions in general. Many of the insights thus derived have closely paralleled the insights developed by clinicians and learning theorists interested in medical education [Elstein, 1986, Donnison-Speijer, 1991, Abel, 1992, Wood, 1992]. One of the fortuitous conceptual developments occurring in the discipline of Artificial Intelligence was that of knowledge engineering. Work in Artificial Intelligence decision support systems in medicine emphasizes the use of problem specific models [Dresdner, 1992]. Artificial Intelligence systems are designed to support the identification of the problem(s). The construction of appropriate models of the problem, the selection and specification of appropriate solution functions, and the identification of significant gaps in knowledge. Since some Artificial Intelligence approaches [Levy, 1979, Levy, 1980, Dojat, 1992, Ruggiero, 1992, Vannobel, 1992] are designed to handle specific kinds of medical problems, one will see considerable difference in emphasis on the various common mechanisms in specific programs.

The general objectives of an Artificial Intelligence program [Rialle, 1991] can be summarized in a few essential points:

- give precedence to a declarative form over any other form of knowledge expression. This principle allows both to easily update knowledge (addition, modification, deletion) and to represent it in natural language or graphic form;
- declare a considerable amount of knowledge required by the Artificial Intelligence system. Indeed, to the basic domain knowledge must be added the

secondary knowledge which is often obvious to the expert;

- explain in plain language all reasonings and inferences.

1.2.5 Knowledge-Based Systems in Medicine

Knowledge-based systems are used for interpretation of data about a specific problem, in the light of knowledge represented in the knowledge base, to develop a problem specific model and then to develop plans for problem solution. Such systems involve the interaction of two components - the descriptive or factual knowledge base itself and specific problem-solving, reasoning or normative knowledge - in addition to client data or patient data characterizing the specific problem [Hirsch, 1992].

The knowledge-base thus has several components [Jackson, 1990]. The knowledge-base proper contains the descriptive or factual knowledge pertaining to the domain of interest. Another portion of the knowledge-base contains the normative problem solving or procedural rules that specify how manipulation of the factual knowledge is to occur as data arrives to characterize the problem in order to reach a decision. There must be a mechanism through which problem data probes the knowledge-base to derive candidate hypotheses therefrom through some pattern matching system, as well as to extract the normative solution functions relevant to the application of these candidate models in the knowledge-base to reach a decision. There is then a reasoning *engine* that carries out the manipulation specified to reach a decision, a decision that may be the final output of the system or an intermediate decision that is fed back to the procedural portion of the system to guide further system activity.

Some of these previously discussed decision support features of an idealized system are illustrated in Fig. 1.3 [Williams, 1982]. Patient data are acquired, either automatically or manually entered or both. Through tactical rules strategic control

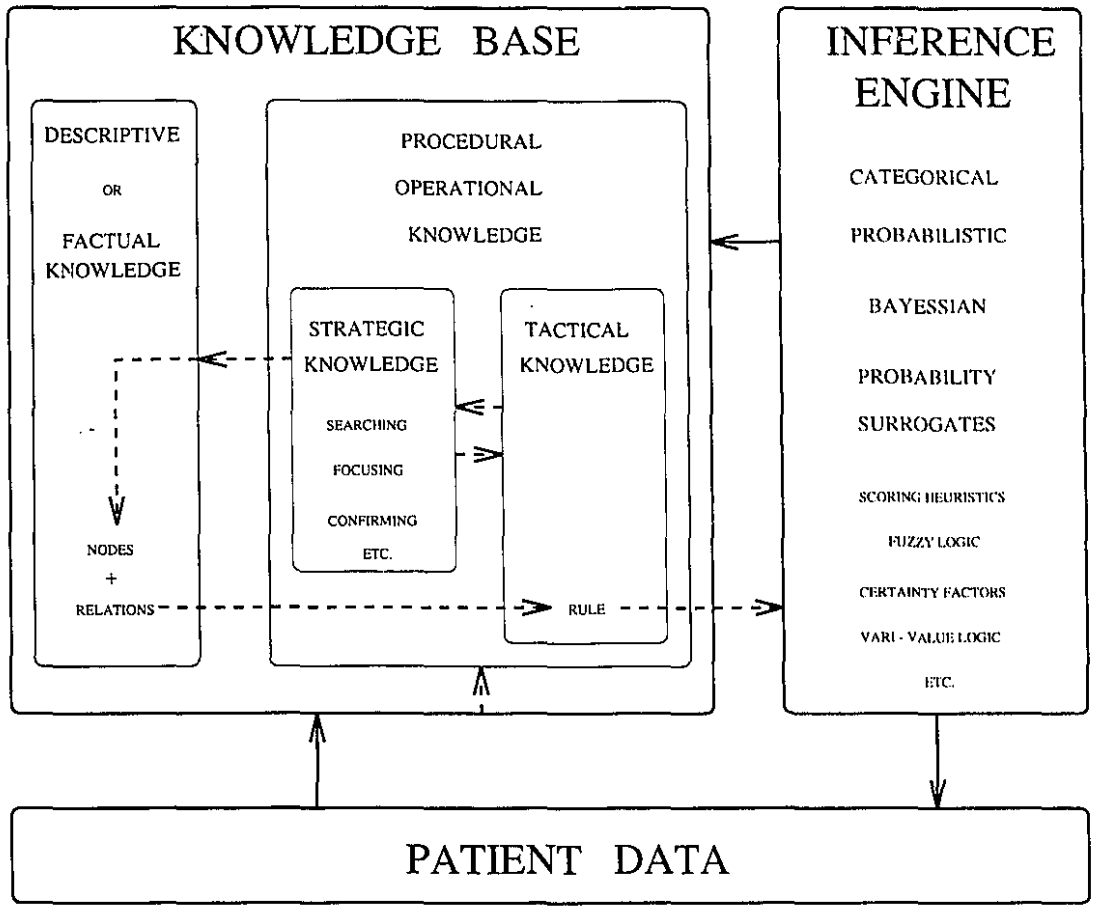

Figure 1.3: A Knowledge-Based System.

is specified for using the patient data to probe the descriptive knowledge-base and activate *facts* relevant to problem solving.

1.2.6 Computer Based Decision Analysis

The volume and complexity of medical knowledge has begun to make computer-based decision-making aids essential to the practice of medicine. Among the various tools and techniques that are currently available, decision analysis provides a particularly promising vehicle for making decisions which involve complex trade-offs [Jordan, 1991]. The technique can be used to arrive at answers to pressing clinical questions, as well as to study the processes of medical reasoning [Sox,

1988]. The process of medical decision making is a broad notion which may involve several steps of diagnosing and medical treatment [Krusinska, 1991].

1.2.7 Interface Devices for Medical Users

An important aspect to be considered is the adoption of new powerful data models that realize a full integration of all the heterogeneous (complex) data of the patient folder. Object oriented multimedia systems appear to be very good candidates for solving the representation problems yet present some open problems with respect to the interaction [Tarantino, 1992].

In object oriented systems links between complex objects are built-in and the interaction is mostly based on the data base navigation. These features give such systems superior consultation capabilities. On the other hand, most object-oriented systems currently offer only limited *ad hoc* query capabilities. A recent approach to object-oriented data bases based on the combination of deductive and object-oriented features a single system, allows one to retain all the modelling and performance advantages of the object-oriented data model while guaranteeing an effective interaction for a variety of users [Staes, 1991].

Multimedia data (graphics, images, sound) enrich the textual content of the document, and might be usefully exploited by the user in the query formulation and by the system in the consequent search process. Given the current nature of multimedia queries, researches on similarity measures for text, speech, graphics and images play an important role in content-based retrieval techniques for multimedia medical information systems. It is extremely helpful that the system be able to rank the retrieved documents according to the probability of the relevance with respect to the user's request. The presentation phase plays a critical role, and the user-system information exchange of this stage must be carefully designed, based on the relevant feedback and incremental query refinement techniques. A proposal

for the retrieval of documents from medical image document bases may be found in [Orphanoudakis, 1991].

Traditional I/O devices do not meet the requirements of medical activities and are not adequate to carry on graphical interactions [D'Atri, 1992]. Research goals are the identification of atomic graphical activities that are relevant for medical applications, the definition of the set of desirable features, and the evolution of criteria for I/O devices and interaction styles. This framework will assist the user interface designer in selecting the appropriate hardware and software tools, by considering the user's capability to carry on a graphical dialogue in an effective, safe and friendly way [Bernabei, 1992]. The graphical display of results forms an important method of visualizing medical data and this topic will be elaborated on in chapter 4.

1.3 Thesis Overview

This thesis applies Artificial Intelligence (AI) techniques for monitoring the condition of inpatients based on information collected by a patient data management system (PDMS) in an Intensive Care Unit (ICU). In this chapter Artificial Intelligence, one of its components (expert systems), and computer-based medical systems have been briefly described. The current version of the PDMS is explained in the following chapter. Chapter three outlines the monitoring system and the algorithms written in C as well as the implementation of the expert system developed in CLIPS 5.0. The Graphical User Interface (GUI) is presented in the fourth chapter. The results are discussed, and future extensions are given in chapter five follow by the conclusion in the last chapter.

Chapter 2 Patient Data Management System

With the hope of improving work efficiency and allowing more nursing time for direct patient care, clinical information systems are becoming more and more popular in clinical environments. This leads to the identification of various issues affecting the acceptance of the systems, ranging from cost considerations to resistance of personnel to using their new computing systems. Many computerized systems are criticized when compared to the old paper-and-pen implementations: difficult to learn, cumbersome in entering data, inability to view different information simultaneously, and so on. The Patient Data Management System (PDMS) is a project developed at the McGill Centre for Intelligent Machines (MCIM) in collaboration with the Pediatric Intensive Care Unit (PICU) of the Montreal Children's Hospital.

The main motivation for developing the PDMS was the need for increased automation in the acquisition and management of patient physiological data, as the monitoring systems at the patient bedside were themselves being automated.

One difficulty with these bedside monitoring systems is the requirement to process a large volume of data generated electronically. Much of the data is presently lost as the health professionals simply take readings of the vital medical and fluid balance data, at the patient's bedside, at certain intervals. These readings are then accumulated and plotted for the interpretation of a patient's state over the past few hours or days. The limitations of the manual data acquisition techniques are as follows: 1) the readings taken at these intervals may not accurately reflect the state of the patient between the readings, and 2) the high probability of errors in transcription, addition, and entry . In addition, if a patient requires resuscitation, then all the data acquisition is usually suspended until the patient's condition sta-

bilizes. Moreover, an unnecessary burden is placed upon the nurses, who would prefer to spend more time on patient care rather than on the manual acquisition and plotting of patient data.

In order to improve the process of manual acquisition, the PDMS was conceived to handle patient data acquired automatically from the bedside physiological monitors, as well as data which cannot be acquired automatically and must be inputted manually. Data that cannot be acquired automatically may include fluid balance data based on the intravenous fluid intake and a variety of fluid outputs for given patient in the intensive care unit. The function of the PDMS can be classified in two main categories. First, to automatically monitor and record the patient data, to facilitate the review and interpretation of the data by presenting colour trends, plots, and charts on a screen display. Secondly, to assist in the hardcopy documentation by producing color printouts of the screens as well as producing the required end-of-shift reports. Since the majority of the data acquired by the PDMS originates from electronic monitoring equipment, the PDMS interfaces directly to the electronic monitoring system; continuously acquiring, logging, and archiving the physiological patient data. The overall motivation of the system is to contribute to a higher standard of patient care by presenting data to the health care professionals which would otherwise be lost in the manual charting method between the charted intervals, as well as relieving the nurses from performing clerical tasks which can easily be done by computer, thus allowing them to spend more time on patient care.

Two main factors must be considered in developing a computer application for a critical care environment such as an intensive care unit. First, the PDMS must be computationally and functionally powerful, while being easy to use for people who are not computer literate. Second, it must meet the needs and expectations of the intended end users to gain acceptance. Finally it must integrate with the existing hospital computing facilities. The following objectives are incorporated into the current version of the PDMS system:

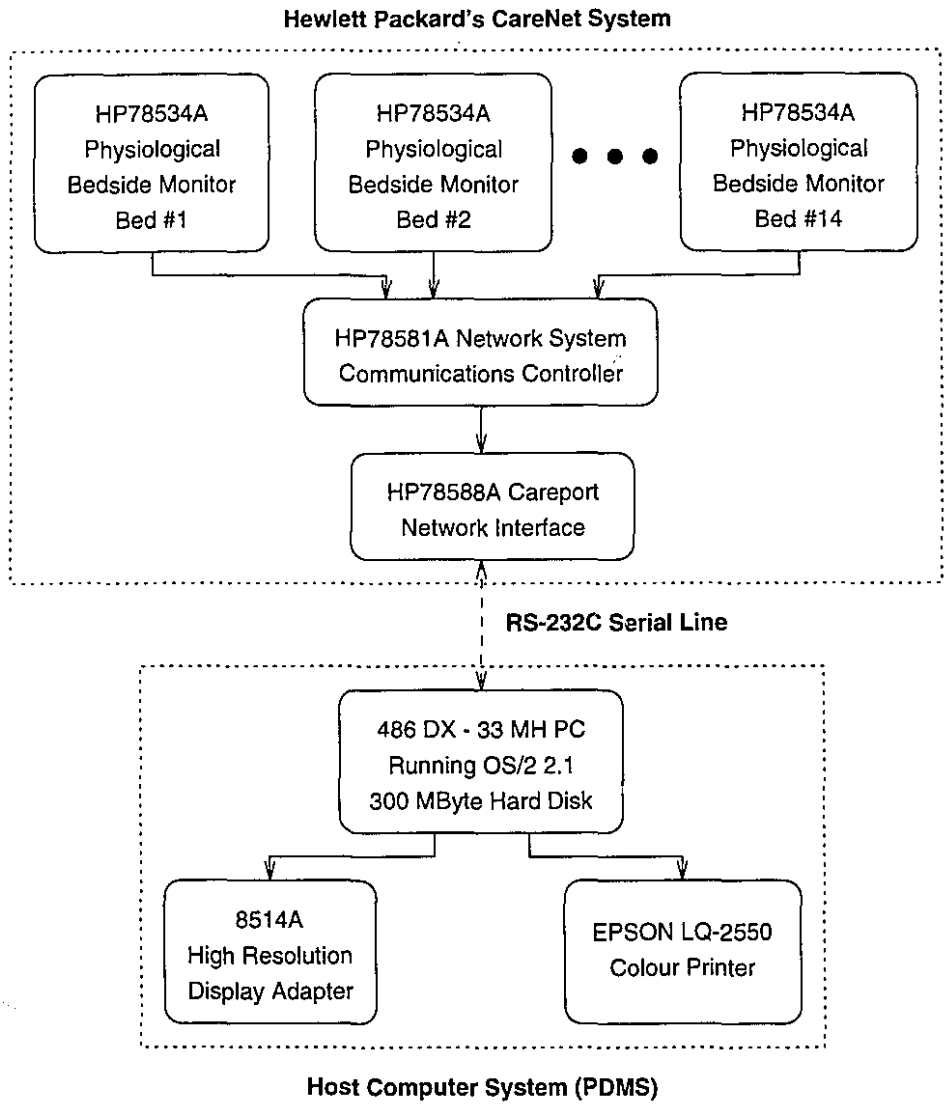

Figure 2.1: PDMS Hardware Platform

- automatically acquiring physiological data from the bedside monitors,
- facilitating the review of the patient data,
- meeting the user's needs for a system which is both powerful and easy to use,
- providing for manual data input in formats which resemble the current paper forms which are familiar to the hospital staff,
- integrating with the existing hospital computer hardware.

2.1 Hardware Platform

The hardware platform of the PDMS, shown in figure 2.1, is based on the Hewlett-Packard CareNet system. The CareNet system provides a local area network which links up HP78532A Physiological Monitor/Terminals located at the patient bedside, to a HP78581A Network Systems Communications Controller in a *star* network architecture. At one of the branches of the star sits the HP78580A Careport. This unit provides a programmable interface between the network controller and a host computer system. The function of Careport is to translate the proprietary network messages and signal formats to standard RS-232C messages which can be understood by the host computer. It also forwards requests for physiological data and other information, to the bedside monitors. In this platform, the host computer system is the PDMS.

Currently, there are fourteen physiological bedside monitors in the Pediatric Intensive Care Unit of the Montreal Children's Hospital. These monitors provide stand-alone data management capabilities which include the real-time display of measured parameters, automatic smoothing of the parameters, and the generation of alarms when either pre-set thresholds are exceeded or when certain critical patient conditions are detected.

The host computer system at the hospital presently consists of a 486 DX - 33 MH personal computer with 16 Megabytes of RAM memory, and a 300 Megabyte hard disk. The display consists of a high resolution color graphics adapter with a resolution of 1024×768 pixels. A colour printer, the EPSON LQ-2550, provides the printout of the PDMS screens in addition to printing out the required forms and reports. A IBM Token-Ring network linking additional PC workstations is planned in the near future.

2.2 Software Environment

The software architecture of the PDMS, a brief outline of some operating system considerations, and the different PDMS software modules are described in this section.

2.2.1 Operating System Considerations

The design of the PDMS has been in progress for some time but the initial design efforts were hampered by the lack of a real-time multi-tasking operating system for IBM personal computers. The release of OS/2 1.0 changed this. The benefits of a multi-tasking operating system enabled the PDMS to be developed as a series of independent, interacting modules to perform the required tasks. This was a clear improvement to designing the PDMS as a single program to perform all of the necessary tasks. Interaction between the modules, which are run as separate processes, is defined in terms of operating system structures such as pipes, semaphores, and shared memory segments. With such a modular design, it then becomes straight forward to add more modules as the design and functionality of the PDMS increases and evolves.

2.2.2 PDMS Modules

In the current version of the PDMS the following modules are implemented:

- The **Startup Module** initiates all other modules, and offers as a main menu selection for allowing the user to select other modules for interaction.
- The **Data Link Controller (DLC) Module** gathers information from the bedside monitor network through a RS-232C link to the Careport network inter-

face. This module has exclusive command over the data acquisition, data storage, and the transmission of commands from other modules to the network. A typical command would be a request for parameters to be transmitted by Careport every n seconds when a patient is admitted to the ward and added onto the PDMS. As this module does not require direct user interaction, it is run in the background.

- The **Patient Registration Module** handles the admission and discharge of patients to and from the PDMS in the Pediatric Intensive Care Unit. This module also handles administrative patient information, such as the patient's name, date of birth, and hospital id number.

- The **Fluid Balance Module** manages the data concerning the fluid intake and output of a given patient. The format of the fluid balance sheet is in a spreadsheet format, effectively emulating the *look and feel* of the paper forms used in the intensive care unit. All fluid balance data must be entered into the PDMS manually.

- The **Trend Display Module** displays the data acquired from the bedside physiological monitors using graphical trends, facilitating the review of the patient's state by the health care professionals. The graphical trend scale can be selected to display data points at one-half hour intervals, minute intervals or second intervals.

The overall organization and interaction between the PDMS software modules listed above is shown in figure 2.2. This organization was partly shaped by the limitations of version 1.0 of the OS/2 operating system. This first version of the operating system supported only text modes of operation, requiring the programmer to implement the graphics routines required in the application program. The modules have subsequently been upgraded to exploit the features of presentation Manager of OS/2 version 1.3.

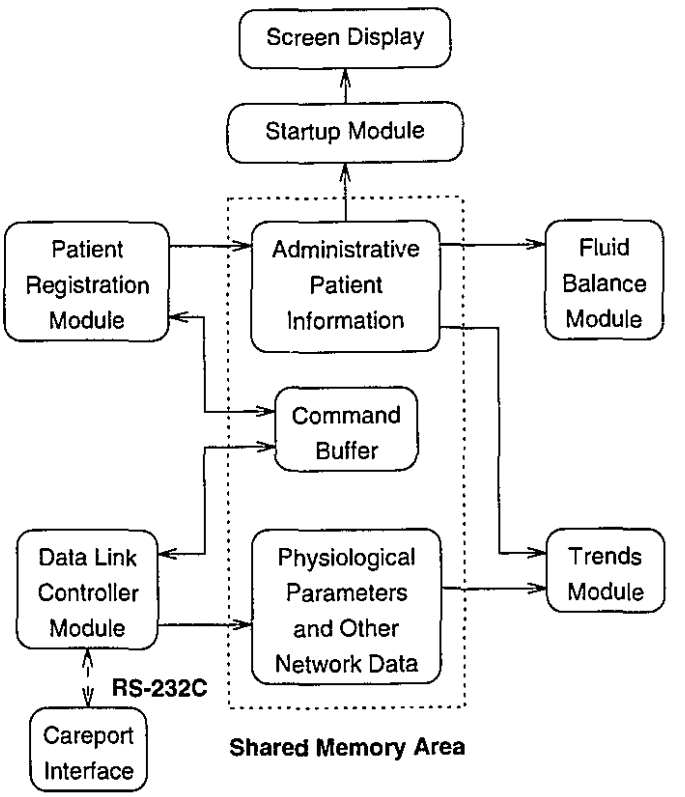

Figure 2.2: PDMS Software Environment

The Startup Module

The Startup module is first PDMS module started, and it, in turn, starts up the other modules as different *child* screen groups. This module then displays the main menu bar of the PDMS, from which any one of the other modules, except the Data Link Controller module, can be selected. When a module is selected by the user, the Startup module opens a full screen window and proceeds to put the selected module in the foreground, allowing user interaction. By selecting the option, one can return to the main menu screen. The modular implementation allows the addition of new modules in the menu bar and simply starting the new process when selected by the user.

2.2.3 The Data Link Controller

The Data Link Controller (DLC) module is responsible for interfacing the PDMS with the bedside monitor network through Careport. DLC stores the information acquired automatically by Careport for easy access by the other modules, and transmits commands received from the other PDMS modules in the proper format to Careport. To do this, DLC exploits OS/2 operating system features such as shared memory segments, pipes, and multiple threads.

The principal function of DLC is to acquire the physiological parameter values from the bedside monitors every two seconds and to place them into circular queues. The *seconds* data is averaged every minute, and these values are placed in minute queues. For each bed, these minute values are in turn averaged every one-half hour and placed in the one-half hour queues. All these queues are located in shared memory, so that these values can later be accessed by the Trends module for graphic display. Periodically, DLC archives data from the minute queues and half-hour queues to the hard drive for future use.

Information about the bed states on the network must also be shared with the Trends and Fluid Balance modules. This information is available to the aforementioned modules through shared memory segments. Access to these segments by the respective processes is controlled by semaphores.

Most of the DLC's activity deals with communicating with the Careport interface. This involves three levels of activity: a subset of the ANSI x3.28-1976 communication protocol which must be respected for physical message transfer, the encoding and decoding of messages from the logical message format of Careport, and the proper definition and manipulation of logical sources in order to obtain the desired network information. Admission, suspension, or discharge of patients on the network is done through the use of logical source definitions. These definitions are virtual connections between the different data sources which are designed to increase the efficiency of the serial communications line, in addition to determining

the manner in which the information is passed to DLC from the network.

DLC automatically receives data sampled every two seconds from the Careport interface. This sampling rate was chosen because it appears to be adequate for present needs and falls within the operational constraints of the equipment available at this time. In addition, alarm messages may be transmitted by Careport to DLC asynchronously, as they occur. The different parameter averaging functions as well as the reception and transmission of data are implemented as different threads in DLC. Communication of relevant information between threads is achieved through pipes.

The Patient Registration Module

This module manages the administrative patient information as well as the admission, suspension, and discharge of patients from the PDMS's representation of the intensive care unit ward. A general menu of commands is presented to the user corresponding to the chosen function. The user can enter, modify, or review both the patient and ward information. This module creates a shared memory structure for storing the administrative patient information, which is indexed by bed number, and is accessible by both the Trends and the Fluid Balance modules so that they can display the patient name with the respective patient data. This module flags events such as the addition, suspension, or discharge of a patient to DLC through the use of shared memory and semaphores.

The Trends and Fluid Balance Modules

The Trends and Fluid Balance modules constitute the most important parts of the PDMS, implementing the main functions for which the PDMS was created. The Trends module allows the review of the physiological parameter values of a particular patient with different horizontal time scale resolutions of seconds,

minutes, or half hours and with adjustable scaling of parameter values on the vertical axis.

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 closely emulates the actual paper forms used in the intensive care unit by the nurses. As previously mentioned, the fluid balance data is still input manually in this module, hence despite computerization, the possibility of data entry errors still exists.

Keyboard bedside data entry for the Fluid Balance module data presents a bottleneck, since there is currently only one PDMS console and fourteen beds in the pediatric ICU. Ideally, the fluid balance data should be read by the nurses, and entered directly into the PDMS. Having one console in the middle of the intensive care unit ward would require nurses to transcribe values read from the infusion pumps onto paper for entry into the PDMS at a later time. This process is unacceptable for two reasons. First, the possibility of transcription errors would still exist, as nurses would still be required to write down data on paper, and later type it into the PDMS. With this system, the nurses' workload would actually increase, and not decrease as desired. 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, while attempting to minimize the occurrence of entry and transcription errors, and facilitating human-computer interaction, data entry using a speech interface was proposed. The interface which would use both speech recognition, for data entry and module operation, and speech generation, for feedback, verification of spoken commands, and audio prompting, providing an *eyes-free* and *hands-free* means of

directly entering fluid balance data at the bedside and at a distance from the PDMS console. Even if computers were introduced at the patient bedside, the speech interface would still be an integral part of the bedside data entry interface, since the intravenous solution infusion pumps can be situated at different locations around a patient's bed. A *hands-free* and *eyes-free* system would permit the nurse to have extra mobility.

Another consideration in the Fluid Balance module of the current PDMS version is the quality of the user interface. As previously mentioned, the Fluid Balance module's interface effectively emulates the spreadsheet format of the manual Fluid Balance sheet. The current Fluid Balance module's Ingesta sheet, for recording fluid intake, and Excreta sheet, for recording fluid output, cannot be displayed completely on one 1024 × 768 pixel display. Consequently, cursor keys and other specially assigned key combinations, such as *ALT-scroll keys* and *CTRL-scroll keys* are used to navigate through the spreadsheet. The key assignments for this module are much more complex than those used in other PDMS modules, presenting an interface which could be quite confusing for a new user. In addition, the movement between the rows and columns on these sheets is quite slow. It typically requires one second to scroll in a new row or a new column into the screen display.

Thus it was decided to implement the next version of the Fluid Balance module using the graphical interface of OS/2's Presentation Manager. All the drawing routines are handled by the operating system, and the graphical user interface is consistent among all applications which use Presentation Manager. The speech interface was then added to the module, adding another input and output modality. The modularity and high-level device independence of the OS/2 system function calls allow alternate input modalities to be added on in a relatively seamless manner.

This chapter has presented the current design of the PDMS which is being developed for the Intensive Care Unit of the Montreal Children's Hospital. The

hardware platform and the software architecture of the PDMS, some operating system considerations, and the various PDMS software modules were described. The following chapter will discuss the development of knowledge-based monitoring system for this environment.

Chapter 3 An Expert System for Monitoring Vital Signs

Understanding the design process has long been a goal of engineers, architects, and others. Such an understanding could lead to better designs, more rapid production of new designs, and greater progress towards meeting real needs and improving our environment. Expert systems provide a new tool for this, by allowing us to express design knowledge in terms that both humans and computers can do somethings with [Rychener, 1988].

One of the expert system programming languages is CLIPS, written in C, and used for real-world projects. CLIPS is a non-procedural, declarative, rule-based programming language. The code developed in CLIPS runs very fast and is designed specifically to provide high portability, low cost, and easy integration with existing or conventional software systems. CLIPS is also designed to interface with other languages such as FORTRAN and Ada.

The basic elements of CLIPS are:

1. fact-list; global memory for data
2. knowledge-base; contains all the rules
3. inference engine; controls overall execution

CLIPS can be called from a procedural language to perform its functions and then return control to the calling program. Procedural code can also be defined as external functions and called from CLIPS. When the external code completes

its task, control returns to CLIPS. CLIPS has a LISP-like syntax which uses parentheses as delimiters. The LISP programming language strongest features include symbolic data representation, dynamic storage allocation (especially for linked list structures), flexible variable binding, the processing of program results by other programs, and the computation of recursive functions required with trees, graphs, and other complex data structures [Rychener, 1988]. Although, CLIPS is not written in LISP, the style of LISP has influenced the development of CLIPS.

A CLIPS program consists of rules, and facts. Facts are very important, because execution in CLIPS can not proceed without them. CLIPS uses a powerful pattern matching known as the Rete Algorithm. The Rete Algorithm is a very fast pattern-matcher that obtains its speed by storing information about the rules in a network. Any C compiler that supports the standard Kernigan and Ritchie C language can be used to install CLIPS. The speed of CLIPS on a 80386 or a 68020 microcomputer is compatible with the state-of-the-art LISP-based development software and LISP machines [Giarratano, 1989].

3.1 Fact-List

In order to solve a problem, a CLIPS program must have data or information with which it can reason. A *chunk* of information in CLIPS is called a *fact*. A fact consists of one or more fields, enclosed in matching left and right parentheses. CLIPS will accept any combination of fields as a fact. Good programming style dictates that a method be used to ensure facts are represented in a meaningful manner. Groups of facts having the same relation can be described for purposes of documentation by using a *template*. A template for a fact shows the relation name followed by one or more general or specific items. More than one template can be used to indicate variations of a set of facts. Templates are convenient for documenting facts and the expected fields that are found in them. They are not accepted by CLIPS as commands and should only be used for documenting CLIPS

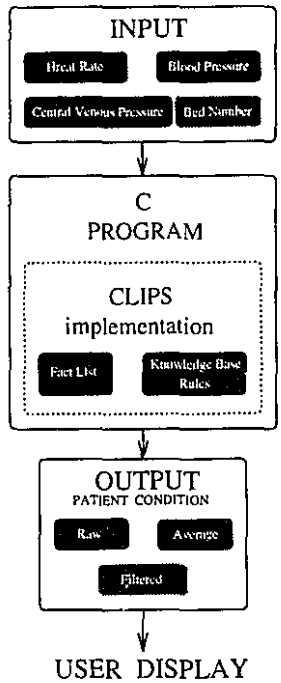

Figure 3.1: The input information of the CLIPS implementation.

code. Further information about CLIPS can be found in [Giarratano, 1989].

We shall now describe the implementation of the vital sign monitoring system carried out using the CLIPS language.

3.2 Input / Output

Figure 3.1presents the information flow of the expert system. The inputs include a *BED NUMBER* plus parameter values for the *HEART-RATE*, the *BLOOD-PRESSURE*, and the *CENTRAL-VENOUS-PRESSURE* of the patient. The OUTPUT of the vital sign monitoring system is a diagnosis of twelve frequently occurring patient conditions for each patient. Three diagnosis interpretations are evaluated namely *RAW*, *AVERAGE*, and *FILTERED*. A C program is used to sequence through the occupied beds accessing the patient vital sign data, performing the averaging and filtering, computations, calling the CLIPS program for evaluating the patient diagnoses. The output patient conditions together with the relevant

bed numbers and a timestamp are saved in a disk file to be used as input by the graphical display program which will be explained in chapter 4.

The RAW_{output} is defined as *the patient's condition based on the latest set of inputs received*. This RAW_{output} is most responsive in indicating a critical situation as it develops. $AVERAGE_{output}$ presents the diagnoses derived from using $AVERAGE_{data}$ vital sign parameters for the heart rate, blood pressure, and central venous pressure, the averaging being evaluated over the previous Number of Data ND. ND has been empirically set to ten to generate a *smoothed* or less noisy value. The $AVERAGE_{data}$ is computed as follows:

Computing $AVERAGE_{data}$
. Define: ND 10
. Define: Vital Sign Parameter VSP_i
. if: $i < ND$
. then: Average of $VSP = \frac{\sum_{n=1}^{i} VSP_n}{i}$
. else: Average of $VSP = \frac{\sum_{n=0}^{9} VSP_{i-n}}{10}$

where i is the current vital sign parameter measured. The $FILTERED_{output}$ presents the percent probabilities of occurrence of each diagnosed patient condition computed as follows:

$$PC_n = \frac{(LPC_n \times (ND - 1)) + (100 \times C_n)}{ND} \tag{3.1}$$

where C_n is 1 if diagnostic condition n is asserted by the Expert System *Medical Rules*, or 0 if this condition is not. PC_n gives the updated percentage probability of occurrence of a condition, LPC_n is the previous percent probability of occurrence value, and ND is a the number of data points in the averaging window. In our implementation ND was chosen as 10 giving.

$$PC_n = 0.9 \times LPC_n + 10 \times C_n \tag{3.2}$$

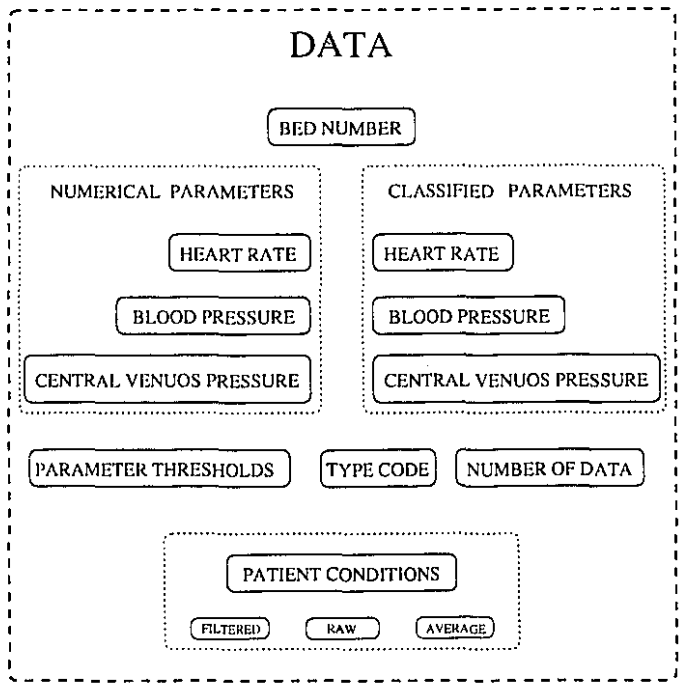

Figure 3.2: The components of data in the program.

The following section presents the data structures used for the algorithm.

3.3 Data Structures

Fig. 3.2 shows the components of the data used in the program namely:

1. Bed Number
2. Numerical parameters for Heart Rate (HR), Blood Pressure (BP), and Central Venous Pressure (CVP).
3. Parameter thresholding values, MIN and MAX used for the classification of HR, BP, and CVP.
4. Classified parameters values for Heart Rate, Blood Pressure, and Central Venous Pressure which can be Low, $Normal$, or $High$.

5. The Type Code is used to reference the two sets of input parameters being processed: the RAW and $AVERAGE$ values.

6. Number of Data (ND) used to compute the parameter $average$ and the $filtered$ diagnosed patient conditions.

7. New sets of output patient conditions computed: raw, averaged and filtered. Each set presents the scoring for each of the twelve possible patient condition diagnoses.

3.3.1 Results and Data-History Files

The output patient condition diagnoses are written in two disk files: a DATA-HISTORY file which is used for testing and debugging purposes, and a RESULTS file which is processed by the graphical user display program. Their structures are shown in section 3.5. We shall now explain portions of the algorithm.

3.4 Components of the Monitoring Algorithm

The implementation of vital sign monitoring program contains of three parts: 1) Data Processing System (DPS), 2) Expert System (ES), and 3) Graphical User Interface (GUI). The Data Processing is written in the C language and is responsible for preparing data for the Expert System and the Graphical User Interface. Figure 3.3 shows the flow chart of the Data Processing System and its interconnection with the Expert System and the Graphical User Interface. The Data Processing System and Expert System processes will presented in the following sections and the Graphical User Interface will explained in the next chapter.

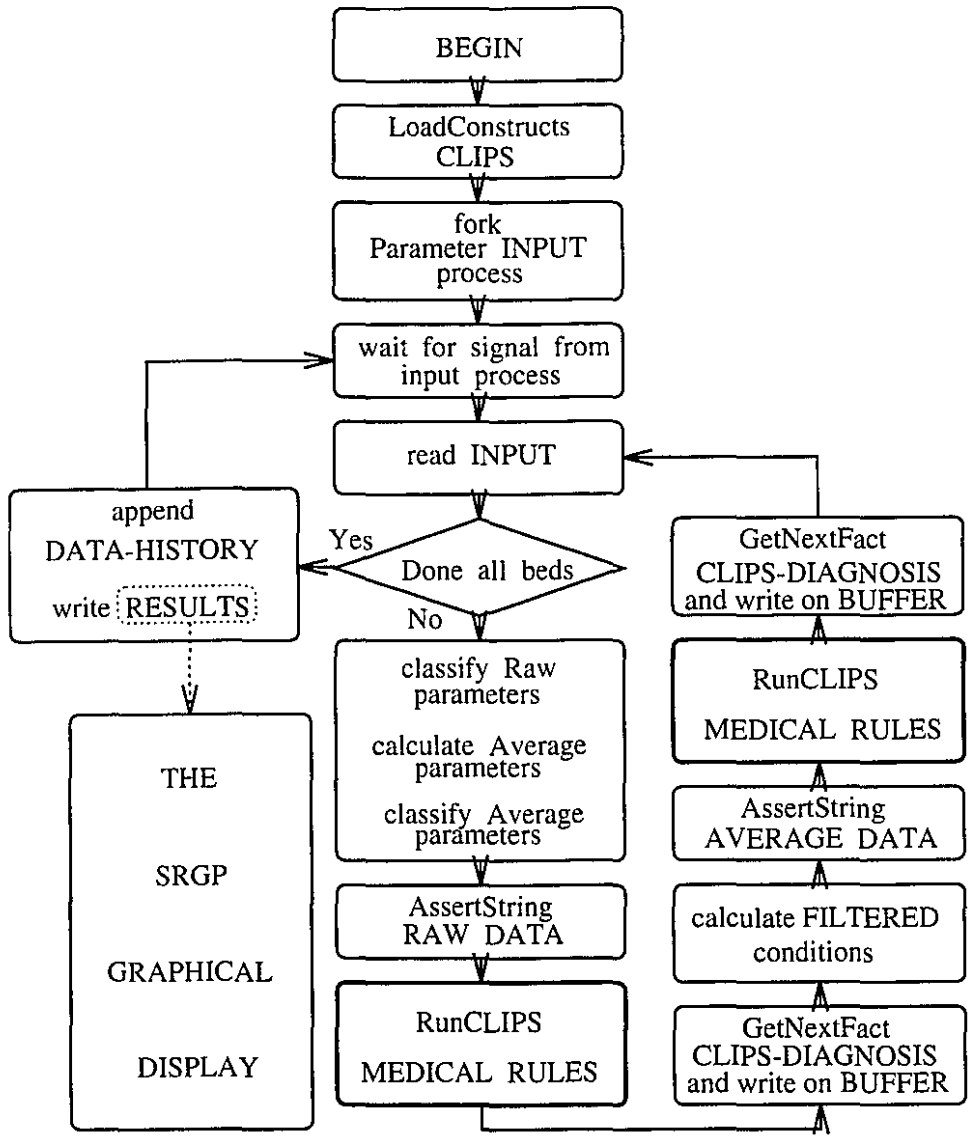

Figure 3.3: Flow chart of the procedure.

3.5 Data Processing System

We shall now outline the Data Processing System and its connection to the Expert System as shown in figure 3.3. The implementation combines both C and CLIPS programs. The procedure begins by loading the CLIPS MEDICAL RULES using the *LoadConstructs* command. The MEDICAL RULES will be explained in section 3.6. Default parameter values and classification threshold values are calculated based on the patient age. The program continues by creating the process which supplies

the new vital sign parameter values and signals their arrival to the vital sign monitoring system with the SIGALARM signal from GETITIMER using the real time counter.

The program waits for new information to arrived. Upon receiving the signal, the process reads in RAW parameter data for a bed. If any parameter values are not supplied, the program will use the default values for the heart rate, blood pressure, or central venous pressure. The program then, classifies the RAW parameter values, computes the average parameter values using the algorithm explained in section 3.2 and then, classifies the $AVERAGE$ parameter values. The parameter values of the RAW and $AVERAGE$ data are classified into the Low, $Normal$, and $high$ conditions using the *CLASSIFY-PARAMETER* procedure. $Normal$ stands for the situation where a parameter lies within the normal range. Low shows a condition below the minimum threshold and $High$ presents a condition above the maximum threshold. These levels are obtained from medical references and vary with patient age.

The program then asserts the facts containing RAW data in the CLIPS agenda using the *AssertString* command. The *RunCLIPS* command begins execution and allows the medical rules to be fired. When CLIPS ends, it returns to the C program where the *GetNextFact* function transfers the CLIPS-DIAGNOSIS results into the buffer of the C program. The next step calculates the $FILTERED$ outputs which give the percent probabilities of each of the twelve possible patient condition diagnoses using equation 3.2. The above sequence (*AssertString, RunCLIPS, and GetNextFact*) is repeated using the $AVERAGE$ parameter values. After processing all the incoming bed data, the final step writes the new information to the RESULTS file for subsequent use by the graphical display program, and appends the new information to the DATA-HISTORY file which records information concerning all the inpatients. The process goes back to the *read* stage to process all the available bed input data and then suspends itself until a new signal arrives. The loop continues until it is terminated by killing the program.

The following pseudo-code summarizes the procedures and data structures of the Data Processing System algorithm:

```
    VITALSIGNS structure
.       BED-NUMBER
.       RAW-HEART-RATE        ; numerical value
.       RAW-BLOOD-PRESSURE
.       RAW-CENTRAL-VENOUS-PRESSURE
.       CLASSIFIED-HEART-RATE      ;low, normal, or average
.       CLASSIFIED-BLOOD-PRESSURE
.       CLASSIFIED-CENTRAL-VENOUS-PRESSURE
.
RESULTS structure
.       VITALSIGNS structure
.       STATUS-OF-BED       ;occupied or empty
.       DIAGNOSIS-FROM-RAW-DATA
.       DIAGNOSIS-FROM-AVERAGE-DATA
.       DIAGNOSIS-FROM-FILTERED-DATA
.
DATA-HISTORY structure
.       date
.       time
.       RESULT structure
.
PROCEDURE CLASSIFY-PARAMETER
.   (char Parameter, char Parameter-Threshold-Range)
.   begin
.       char Status
.       compute Normal, Low, and High of Parameter
.           if Parameter ∈ Parameter-Threshold-Range
.               then Status is Normal.
.           if Parameter is less than MIN-Parameter
.               then Status is Low.
.           if Parameter is greater than MAX-Parameter
.               then Status is High
.       return Status
.   end
.
.
PROCEDURE
.   begin
.       char BUFFER
```

```
.       LoadConstructs
EXPERT SYSTEM of MEDICAL RULES
.       calculate default parameter values and classification thresholds
.       fork INPUT PARAMETERS
WAIT:
.       wait for SIGNAL announcing new data
READ:
.       read RAW VITALSIGNS input data
.       if no more bed data goto WRITE
.       Classify RAW parameters
.       compute average values of parameter patterns
.       for RAW and AVERAGE parameters
.               CLASSIFY-PARAMETER
.                 (HEART-RATE, HEART-RATE-Threshold-Range)
.               CLASSIFY-PARAMETER
.                 (BLOOD-PRESSURE, BLOOD-PRESSURE-Threshold-Range)
.               CLASSIFY-PARAMETER
.                 (CENTRAL-VENOUS-PRESSURE,
.                 CENTRAL-VENOUS-PRESSURE-Threshold-Range)
.           AssertString RAW-DATA
.           RunCLIPS MEDICAL RULES
.           GetNextFact CLIPS-DIAGNOSIS
.           compute probabilities of each condition (FILTERED)
.           write RAW-DATA on BUFFER
.           append CLIPS-DIAGNOSIS for RAW-DATA on BUFFER
.           AssertString AVERAGE-DATA
.           RunCLIPS MEDICAL RULES
.           GetNextFact CLIPS-DIAGNOSIS
.           append AVERAGE-DATA on BUFFER
.           append CLIPS-DIAGNOSIS for AVERAGE-DATA on BUFFER
.           append the probabilities on BUFFER
.       goto READ
WRITE:
.       write BUFFER on RESULTS file
.       append Date, time, and BUFFER on DATA-HISTORY file
.       goto WAIT
.   end
```

3.6 Rule-Based Knowledge

The knowledge for the rule-based monitoring system was obtained from medical publications and from consultations with the medical and nursing experts in the Pediatric Intensive Care unit at the Montreal Children hospital. From this knowledge a medical diagnostic model was formulated relating the classified conditions of the heart rate (HR), blood pressure (BP), and central venous pressure (CVP) to eleven frequently recurring patient conditions as well as a twelfth normal patient state. The monitored conditions are the following:

1. Hypovolemia

   Blood volume is too low either through dehydration, hemorrhage, or extensive burns. Although possibly compensated by vasoconstriction, hypovolemia will reduce the amount of blood coming back in the veins and will ultimately cause the blood pressure to fall since the heart will not be sufficiently filled.

2. Hypervolemia

   Too much blood volume can be achieved if the amount of drug solution is not carefully monitored. Usually the situation is temporary since the excess fluid is removed through the kidneys.

3. Bradyarrhythmia

   The heart beats at an abnormally low rate. Blood pressure will drop and venous pressure will rise with the decrease of flow passing through the heart.

4. Tachyarrhythmia

   The heart beats at an abnormally high rate. At a high enough rate, the ventricles do not have enough time to fill up completely, resulting again in a decrease of blood flow.

5. Tamponade

   This condition is found when a hemorrhage takes place around the heart and

reduces the space needed for it to fill up. The stroke volume is thus reduced, reducing again the blood flow. This is more common for patients with heart surgery.

6. Pump Failure

   This condition describes the failure of the heart muscle to pump blood correctly, usually due to a lack of good perfusion.

7. CNS-ICP

   All the CNS (Central Nervous System) dysfunction label represents some kind of problem in the nervous regulation of the CVS (Central Venous System). An increased Intra-Cranial Pressure (ICP) in the brain affects the nervous centers and causes periods of high vasoconstriction and blood pressure.

8. CNS-Drugs

   The abnormal behavior of the nervous regulation is more probably caused by some inhibitory effect of drugs given to the patient such as sedatives.

9. Primary Hypertension

   As it names describes, this stands for a high blood pressure. It is primary since it is the cause of the other CVS changes.

10. Systemic Shock

    Systemic shock occurs when some chemical affects the vasodilator receptors of the CVS, resulting in a sudden and massive decrease in blood pressure. This chemical can come from bacterial infections or simply an allergic overreaction. It usually starts with a flush of the skin and its progressive cooling.

11. Agitation

    This describes generally the state of a patient under some stress, either from pain, crying or simply restlessness.

12. Normal

    This condition corresponds to the patient having vital sign parameters within

```
(VITASIGNS
    (field HEART-RATE                        ; Low, Normal, High
    (type WORD))
    (field BLOOD-PRESSURE                    ; Low, Normal, High
    (type WORD))
    (field CENTRAL-VENOUS-PRESSURE           ; Low, Normal, High
    (type WORD))
)
```

Figure 3.4: The CLIPS structure of VITALSIGNS.

their normal range.

The patient HR, BP, and CVP vital sign are first classified into *low*, *normal*, or *high* states. Then twenty seven medical rules were formulated to relate these patient's measured states to each of the twelve possible medical conditions listed above. These rules are summarized in tables 3.1, 3.2, and 3.3. Here, the -, =, and + symbols correspond to *low*, *normal*, and *high* parameter values. The asterisk * indicate the medical conditions diagnosed by the rules. For example in table 3.1, the first medical rule states that a patient with a low HR (-), low BP (-), and low CVP (-) is diagnosed to have the following medical conditions (Codes): Hypovolemia (1), Bradyarrithmia (3), and Pump Failure (6), as indicated by the presence of the asterisks against each of these conditions in the first column of the tables. Blank entries imply that those corresponding medical conditions are not supported by the vital sign states of that column. The twenty seven CLIPS medical rules have no priority between themselves. They are applied to process RAW and AVERAGE parameter data sets.

Figure 3.4 shows the structure of the VITALSIGNS template used in the CLIPS program while figure 3.5 presents the CLIPS coding of Medical Rule #1. Tables 3.1 to 3.3 exhibits how the results are obtained. The signs +, -, and = in the tables represent *high*, *low*, and *normal* values; * signifies that the given condition can be present for that pattern of parameters.

	MEDICAL RULES:	1	2	3	4	5	6	7	8	9
HR	Heart Rate	-	-	-	-	-	-	-	-	-
BP	Blood Pressure	-	-	-	=	=	=	+	+	+
CVP	Central Venous Pressure	-	=	+	-	=	+	-	=	+
CODE: 1	Hypovolemia	*			*		*	*		
2	Hypervolemia									*
3	Bradyarrhythmia	*	*	*	*	*	*			
4	Tachyarrhythmia									
5	Tamponade			*						
6	Pump Failure	*		*						
7	CNS-ICP							*	*	*
8	CNS-Drugs									
9	Primary Hypertension							*	*	*
10	Systemic Shock									
11	Agitation									

Table 3.1: Parameter patterns and corresponding patient's condition for *Low* Heart rate (part 1).

	MEDICAL RULES:	10	11	12	13	14	15	16	17	18
HR	Heart Rate	=	=	=	=	=	=	=	=	=
BP	Blood Pressure	-	-	-	=	=	=	+	+	+
CVP	Central Venous Pressure	-	=	+	-	=	+	-	=	+
CODE: 1	Hypovolemia	*			*			*		
2	Hypervolemia						*			
3	Bradyarrhythmia									
4	Tachyarrhythmia									
5	Tamponade									
6	Pump Failure	*		*						
7	CNS-ICP									
8	CNS-Drugs	*	*	*				*	*	*
9	Primary Hypertension								*	*
10	Systemic Shock		*	*						
11	Agitation									

Table 3.2: Parameter patterns and corresponding patient's condition for *Normal* Heart rate (part 2).

```
(defrule HRL-BPL-CVPL                          ; RULE  NUMBER 1
                                                 ; IF
    (VITALSIGNS                                    ; LEFT-HAND-SIDE
                                                     ; FACT
                (HEART-RATE                    LOW) ; LOW
                (BLOOD-PRESSURE                LOW) ; LOW
                (CENTRAL-VENOUS-PRESSURE  LOW)  ; LOW
    )
=>                                               ; THEN
   (assert                                         ; RIGHT-HAND-SIDE
            (CLIPS-DIAGNOSIS  HYPOVOLEMIA)       ; assert
            (CLIPS-DIAGNOSIS  BRADYARRHYTMIA)
            (CLIPS-DIAGNOSIS  PUMP-FAILURE)      ; PATIENT CONDITIONS
   )
)
```

Figure 3.5: The CLIPS coding of MEDICAL RULE #1.

For example, the expert system medical rule #1 corresponds to a *Low* status of all three parameters and implements following algorithm:

MEDICAL RULE # 1

If

. VITALSIGNS with
. HEART-RATE equals to *Low*
. and BLOOD-PRESSURE equals to *Low*
. and CENTRAL-VENOUS-PRESSURE equals to *Low*

Then

. assert:
. the patient's condition codes according to
. Tables 3.1 to 3.3 to obtain
. Hypovolemia
. Bradyarrhythmia
. Pump Failure

The rule asserts Hypovolemia, Bradyarrhythmia, and Pump Failure as facts in the agenda of CLIPS. These are retrieved using the *GetNextFact* function and stored in the BUFFER of the Data Processing System before a new set of parameter values are submitted for diagnosis by the medical rule system.

	MEDICAL RULES:	19	20	21	22	23	24	25	26	27
HR	Heart Rate	+	+	+	+	+	+	+	+	+
BP	Blood Pressure	-	-	-	=	=	=	+	+	+
CVP	Central Venous Pressure	-	=	+	-	=	+	-	=	+
CODE: 1	Hypovolemia	*			*			*	*	
2	Hypervolemia						*			*
3	Bradyarrhythmia									
4	Tachyarrhythmia	*	*	*		*	*			
5	Tamponade									
6	Pump Failure			*						
7	CNS-ICP							*	*	*
8	CNS-Drugs									
9	Primary Hypertension								*	
10	Systemic Shock	*	*	*	*		*			
11	Agitation				*	*	*	*	*	*

Table 3.3: Parameter patterns and corresponding patient's condition for *High* Heart rate (part 3).

3.7 Summary

The fact-list of the algorithm in the knowledge-based system includes the heart rate, the blood pressure, and the central venous pressure of inpatients. The rule-based system comprises twenty seven CLIPS rules. The output of the algorithm expresses the inpatients' medical conditions. The Data Processing System and the Expert System portions of the program and its elements were outlined in this chapter. A discussion on the Graphical User Interface, its implementation issues, and its execution are presented in the following chapter.

Chapter 4 Graphical User Interface

The user interface is considered to be one of the important factors influencing the acceptance of the clinical computer systems by their users. Information representation, interaction style, and interaction environment are some of the important reasons for increasing the usage of graphics in medical information systems [Bernabei, 1992]. Introduction of computerized systems in the ICU environment poses more challenges than in any other clinical settings [Jiang, 1991]. Care-givers in the ICU usually work under greater pressure and may not have much time to spend in getting familiar with new, sophisticated equipment. The computerized ICU system should have a natural interface, making it easy for users to operate. The huge amount of information generated through intensive care monitoring of very critically ill patients and the heavy usage of various patient data calls for a natural and logical representation of that information for direct patient care at the bedside.

4.1 The Simple Raster Graphics Package (SRGP)

The Simple Raster Graphics Package (SRGP) [Foley, 1990] is a device-independent graphics package that exploits raster capabilities. SRGP's repertoire of primitives (lines, rectangles, circles, ellipses, and text strings) is similar to that of the popular Macintosh QuickDraw raster package and that of the Xlib package of the X Window System. On the other hand, its interaction-handling features are a subset of those of SPHIGS, the higher level graphics package for displaying 3D primitives.

We start our discussion of SRGP by examining the operations that applications perform in order to draw on the screen: the specifications of the primitives and of the attributes that affects their image.

Although our discussion of SRGP assumes that it controls the entire screen, the package has been designed to run in window environments, in which case it controls the interior of a window as though it were a single screen. The application programmer therefore does not need to be concerned about the details of running under control of a window manager.

SRGP supports a basic collection of primitives: lines, polygons, circles, ellipses, and text. To specify a primitive, the application sends the coordinates defining the primitive's shape to the appropriate SRGP primitive generator procedure. It is legal for the specified point to lie outside the screen's bounded rectangular area; of course, only those portions of a primitive that lie inside the screen bounds will be visible. SRGP also offer a variety of line styles such as dotted, dashed, solid and with varying thickness choices.

4.2 The Display Program

The objectives selected in designing the user interface for the bedside ICU display system include:

- ease of use by both novice and expert computer users;
- ability to display different patient information simultaneously;
- comprehensive display of all information on one window;
- prevention of visual errors by using redundancy;
- good ergonomic features for working with the system.

These characteristics will make users feel comfortable, minimize or eliminate users' resistance to the computerized system, and help users in their decision making for patient care.

The SRGP software version 1.0 was used to implement the PDMS user interface on a SUN SPARC workstation 1+, model GDM-1604B15 runing BSD UNIX version 4.2.

Figures 4.1 shows the flow chart of the SRGP program. The program starts by opening an X window and follows by creating the graphical frame for the window. Display function of the program converts the input data obtained the RESULT file to meaningful color boxes on the window. Then the control sequence function allows the program to wait until SIGALRM signals that new results have arrived or until the sleeping time[1] which is two minutes ends. Alternatively the signal could come from the Data Processing System after updating the RESULT disk file. In our design, we linked the data flow between the CLIPS and SRGP procedures by disk files making it possible to run multiple copies of the display procedure over a network when supporting multiple concurrent viewers. Section 4.4 will present the techniques used for synchronization of the Data Processing System, CLIPS program, and SRGP procedures. The following algorithm summarizes the graphical display program which was written in ANSI C/SRGP.

Algorithm of Graphical User Interface for the Vital Sign Monitoring System

PROCEDURE Begin ICU Display

```
.    Begin:
.        Define name of window
.        Define size of window
.        Set keyboard processing mode (RAW)
.        Set input mode (KEYBOARD, EVENT)
.        Select Display for Color and Mono monitors
.        If Color monitor
.            load common colors from 0 to 14
.    Make Frame
.            Set foreground and background colors
.            Set line width
.            Draw grid
.            Write text and background headers
.      Loop
.        Begin:
```

[1]This term refers to the suspended state of a process resulting from the UNIX system call.

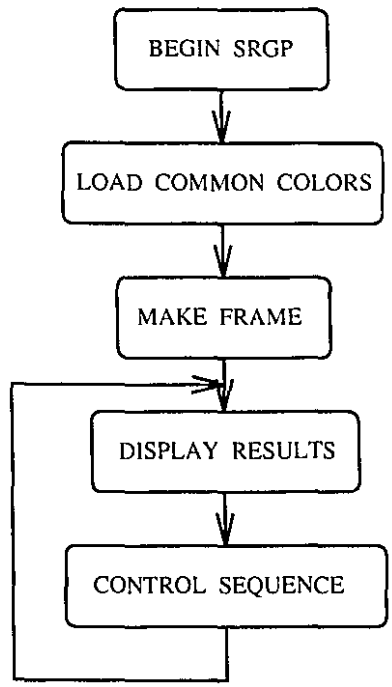

Figure 4.1: Flow Chart of the Graphical Display

```
                Refresh SRGP
                Read results
                Display the results
                Sleep 120 seconds or less if a signal is received
                Go to Begin of the Loop
            End:
        End:
```

Figures 4.2 shows the presentation of the results obtained from the expert monitoring system in the form of a colored table. The color table displays eleven sets of data corresponding to the eleven available diagnoses as well as a twelfth "*Normal*" condition for each bed. Data are organized in groups, and different colors are used consistently to clarify the presentation of results. Fourteen rows display the results for each of the fourteen beds. The patient data and their interpretations by the expert monitoring system are updated automatically. Two alternative displays were developed as shown in figures 4.3 and 4.4. Because of the increased resolution and better detail, monochrome screen display printouts are presented instead of color

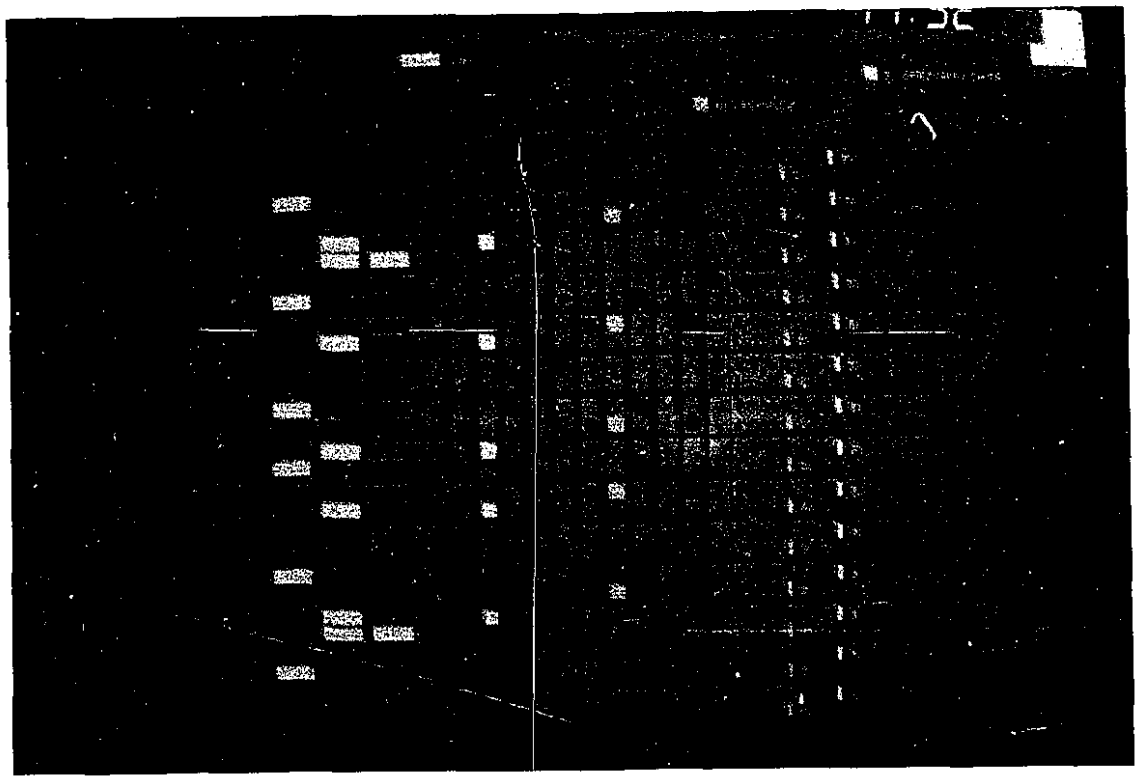

Figure 4.2: Color Print of the Expert Monitoring System Display.

prints. We shall now discuss these display windows in greater detail.

4.3 The Monitoring System Display Screens

The goal of this program is to display the results inferred by an expert system for a physician or a nurse. A total of fifteen colors have been used in the design of these displays. In the color assignments, green signifies a normal condition, red denotes on alarming condition, whereas yellow corresponds to a critical one. The top of Figure 4.2 presents the colored patches and written text giving the color assignments selected for each of the medical conditions.

The table is divided into fourteen rows, one for each patient or bed. The cyan

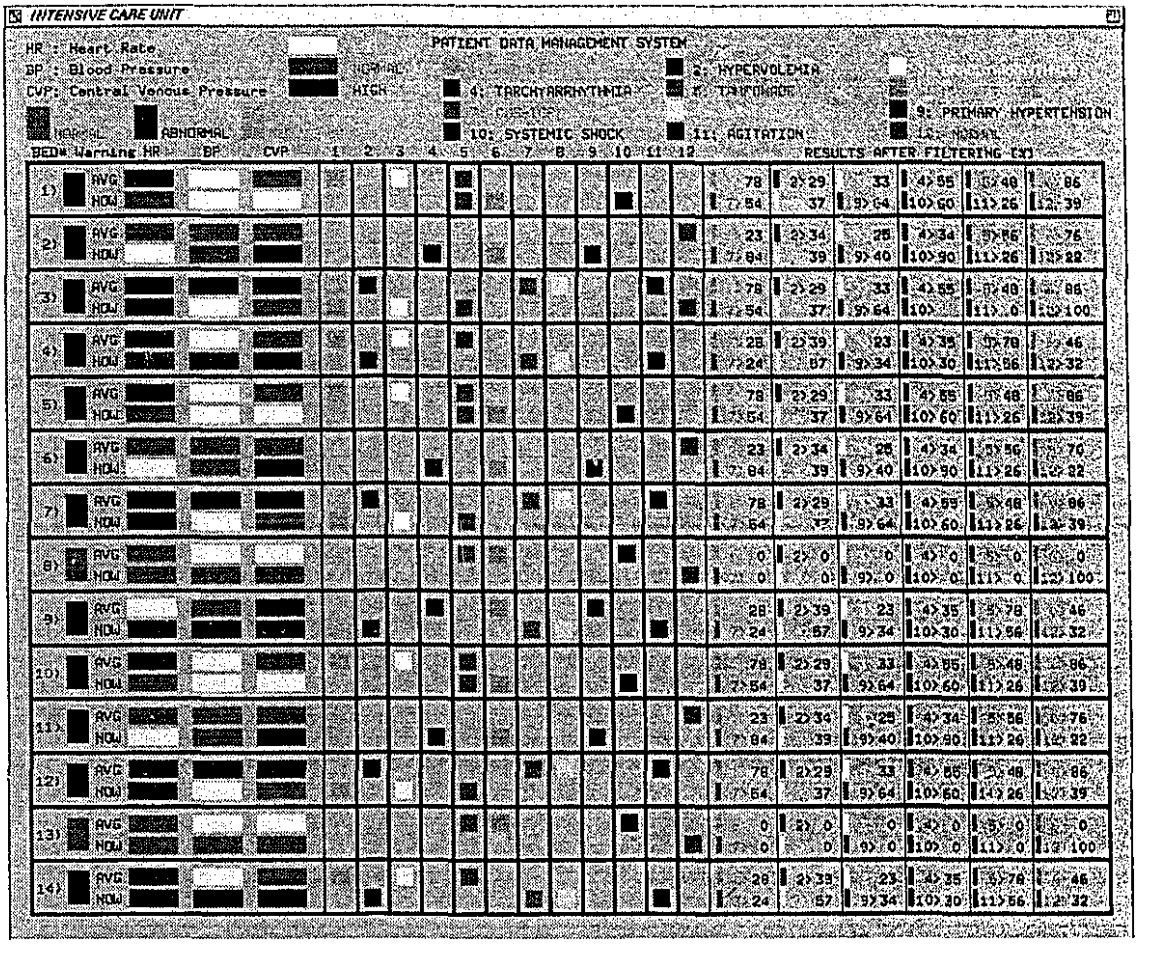

Figure 4.3: Printer Screen Dump of the Expert Monitoring System Display.

color is used to indicate an empty bed condition. Data and information concerning each patient is presented using numbers and colored squares. To facilitate the grouping of information, the window is divided into three parts from left to right (figures 4.2 and 4.3). The leftmost portion, presents the patient state as well as the input vital sign data. The central portion presents medical conditions diagnosed by the expert system on two rows using their relevant colors. The top row shows the interpretation of the RAW values which the bottom row display the $AVERAGE$ results. Finally, the right portion displays the $FILTERED$ numerical scoring for each of the twelve possible patient states. In Figure 4.4, the right hand column

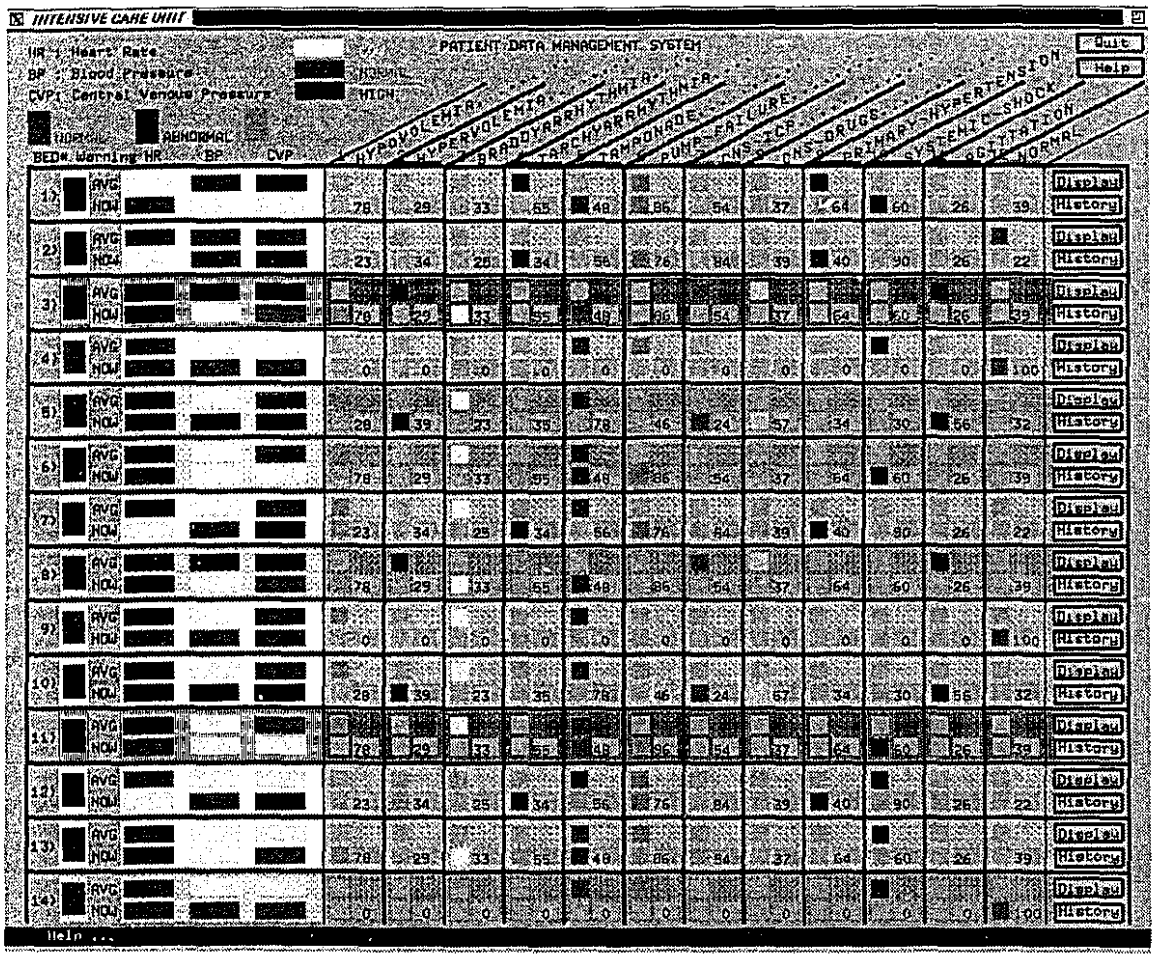

Figure 4.4: Alternative Display for the Expert Monitoring System.

for each bed contains two buttons[2]: *HISTORY* to access the patient conditions recorded in the DATA HISTORY file for review and is not currently implemented; and *DISPLAY* to highlight a specific row or to dim the other rows. Highlighting is performed by making the row background brighter while making the other rows darker, or vice versa. Background color changes can also be used for highlighting depending on the mouse button that is used. At the top of the table, the names of the medical conditions as well as overall patient condition and vital sign color settings appear. These colors are duplicated in each row as small a square which is lit up whenever that condition is present. Therefore, the less color boxes seen

[2]A button is a sensitive area of a display for use in controlling or interacting with a program.

in a row, the fewer the medical conditions afflicting the patient. A help and a quit button are located at the upper right corner. The bottom line of the window is used to confirm the functions invoked by the interface. The display interface was designed to be operated by using a mouse locator but it can easily be extended to include key board control using arrow keys and sensitized keys.

Figures 4.2, 4.3 and 4.4 show the Graphical User Interfaces displaying artificial test data generated using a SHELL program.

4.4 Program Control

The UNIX system manages multiple concurrent processes. A process is the execution of a program. A central part of UNIX or the kernel deals with the control of processes. The kernel schedules the CPU by switching execution from one process to the next. A process can go through a number of execution states before completion, such as, *running*, *asleep*, *ready*, and *zombie*. More detail about the states can be found by consulting the on-line "man" pages of a UNIX environment.

We shall explain some UNIX commands before our process control methods are presented.

- Fork

  A process can create another process via the *fork* system call. The process that invokes *fork* is the *parent* process, and the newly created process is the *child* process. After the *fork* call, the child process and the parents process run concurrently.

- Sleep

  The *sleep* system call allows a process to be paused or suspended for a specified amount of time.

- Interrupts and Signal

  Events outside a process can affect process execution. The time when such an event would occur is not predictable. Thus, they are called *asynchronous events*. *Asynchronous events* are treated in UNIX using the *signal* mechanism. One of the *signals* in UNIX is $SIGALRM$ which is generated by *getitimer* whenever the real timing interval expires.

Instead of using pipes, the results of the Data Processing System are saved in disk files for simplicity and because the disk file is required for testing and possibly for archiving. An input process supplies the vital sign input data to the Expert System process. The process signals the arrival of new data using $SIGALRM$. Combination of the $SIGALRM$ and the *sleep* system call allows the Expert System to wait for the new data. This signalling technique was selected for synchronization of the Vital Sign input process with the Expert System and the graphical displays.

A third interface together with a complementary expert monitoring system was also developed on a NeXT computer with the NeXTSTEP 2.1 operating system, the NeXT Interface Builder and the Objective C Object Oriented Programming Language. Although this development environment was powerful and easy to operate, the voice recognition interface implementation which was also included in this design made it unacceptably slow, typically three minutes per spoken word and it was rejected.

4.5 Summary

A Graphical User Interface (GUI) has been presented in this chapter. The user interface is considered to be one of the important factors influencing the acceptance of the clinical computer systems by their users. The usage of graphics in medical information systems is increasing for several reasons which were mentioned. The GUI is written in C language. Implementation, execution, results, and future

research directions are discussed in the subsequent chapter.

Chapter 5 Implementation, Execution, Results, and Future Directions

5.1 Introduction

As explained, the algorithm is implemented in CLIPS 5.0 and was developed on a SUN SPARC 1 system, under UNIX/X Windows. This UNIX platform was selected because of its availability and its superiority for development and testing compared with the OS/2 platform presently used in the Intensive Care unit. The CLIPS programs are highly portable and can easily be recompiled for the OS/2 environment when necessary. Unfortunately, the SRGP display program does not offer this portability. However the feasibility of adapting the existing OS/2 implementation to a UNIX environment was studied and was found to be possible. After a brief description of the linear regression analysis, the experimental results and directions for future research are presented in this chapter.

5.2 Linear Regression Analysis

Very often the statistician or data analyst is asked whether there is any relationship between several sets of data and then, if there appears to be a relationship, what is the form of the relationship. The simplest relationship is linear. Existence of such a relationship is generally tested by calculating the *linear correlation coefficient* which can be only done for paired sets of data. The correlation coefficient between

a set of data X_i and Y_i ($i \in [1, n]$) is usually denoted by r and given by [Furgeson, 1959].

$$r = \frac{n \times \sum xy - \sum x \times \sum y}{\sqrt{[n \times \sum x^2 - (\sum x)^2][n \times \sum y^2 - (\sum y)^2]}} \tag{5.1}$$

If there is a reasonably strong linear relationship between two sets of data ($r > |0.7|$), then it is justified to calculate what the relationship is. The equation of the regression line is normally expressed in the following form;

$$y = A + B \times x \tag{5.2}$$

where

$$B = \frac{n \times \sum xy - \sum x \times \sum y}{n \times \sum x^2 - (\sum x)^2} \tag{5.3}$$

$$A = \frac{\sum y - B \times \sum x}{n} \tag{5.4}$$

and n is the number of data.

5.3 Experimental Results

The results analyzed in this section were derived from an intermediate design having all fourteen beds, the data processing and the control system implemented in the CLIPS program. Analyzing these results obtained led us to the improved implementation presented in chapter 3. The execution times presented here were derived using the $WATCHALL$ debugging environment in CLIPS and thus represent only a comparative evaluation since they include an execution overhead for the debugger and printing to the screen. Execution without the debugging environment is much faster but the timing observation are more difficult to obtain under those conditions. Thirty eight test runs of the program were executed with different number of input FACTS changing as shown in Table 5.1 using the CLIPS debugging environment. Here, a FACT represents the set of data corresponding

Set	1	2	3	4	5	6	7	8	9	10
Facts	1	1	1	14	14	14	7	3	3	3
Number of Runs	4	6	4	1	abort	abort	3	4	8	8

Table 5.1: Execution tests

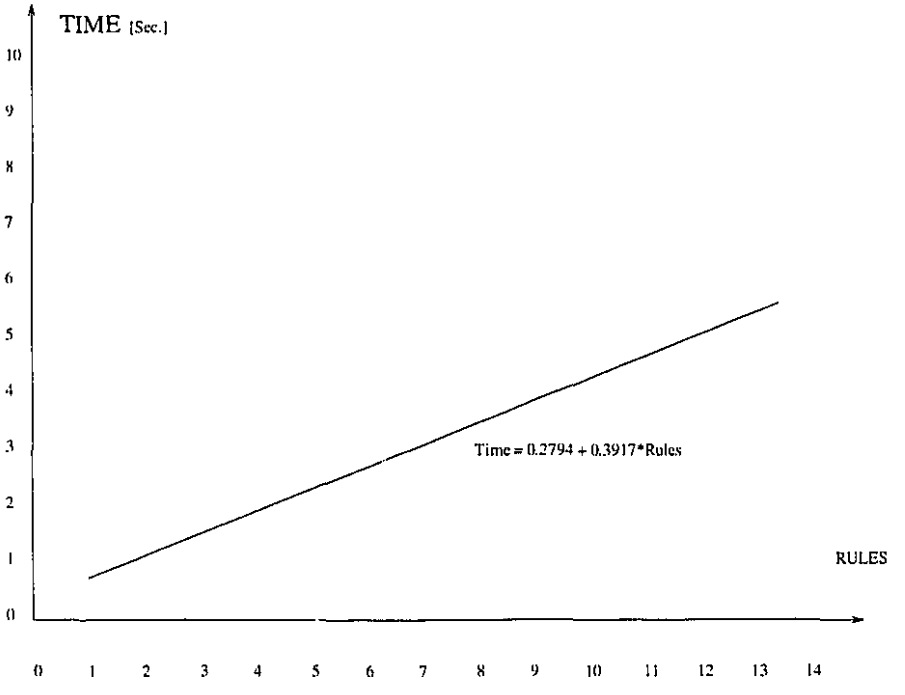

Figure 5.1: The relationship between TIME and RULES for one bed.

to one bed. These test runs were conducted to study the effect of changing the number of FACTS processed by the program. Sets 5 and 6 were aborted and could not complete. The run times are shown on the Table 5.2.

The curves in figures. 5.1 to 5.3 show relationship between run time and the number of rules fired derived using regression analysis on the results given in table 5.2. Linearity was confirmed on the basis of the r statistics shown in Tables 5.3 and 5.4. Here A and B have already been defined, n is the number of samples or runs, T_{avg} and R_{avg} represent average time and rules computed over the sets,

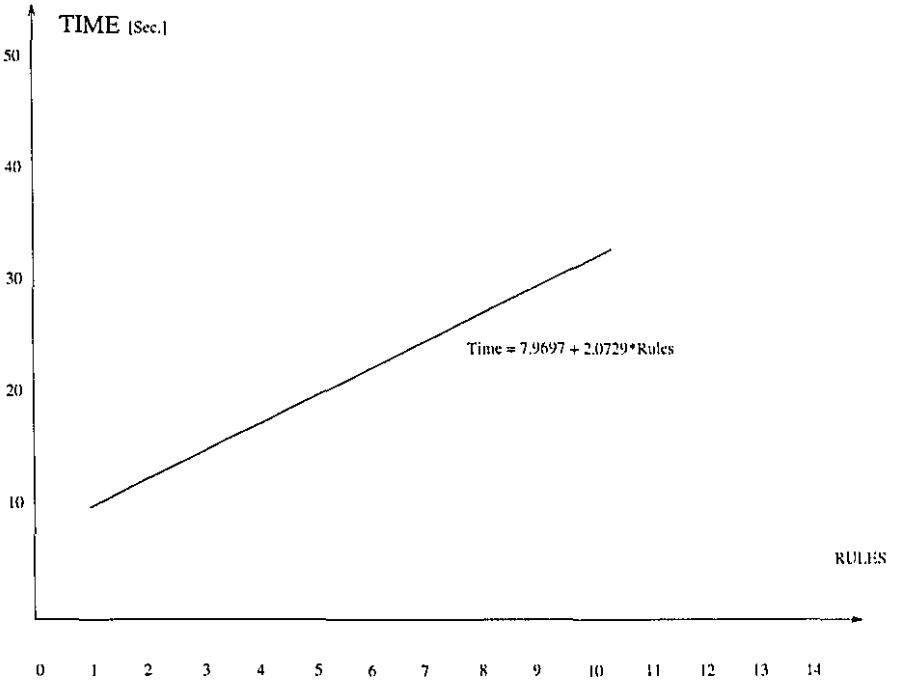

Figure 5.2: The relationship between TIME and RULES for seven beds.

and $\sigma(n)$ and $\sigma(n-1)$ represent standard deviations. As the plots in Figs. 5.1 to 5.3 show, the number of beds is very important in controlling the "speed" of the program ($\frac{\partial rules}{\partial time}$). The equations number (5.5) to (5.11) are linear regression equations summarized in the tables 5.3 and 5.4. Because all the linear correlation coefficients r are greater than $|0.7|$ linearity is confirmed for the regression equations expressing;

$$TIME = A + B \times RULES \tag{5.5}$$

For 3 beds in table 5.3

set 8 gives:

$$TIME = -2.164 + 1.007 \times RULES \tag{5.6}$$

set 9 gives:

$$TIME = 7.789 + 1.186 \times RULES \tag{5.7}$$

set 10 gives:

$$TIME = 1.564 + 0.254 \times RULES \tag{5.8}$$

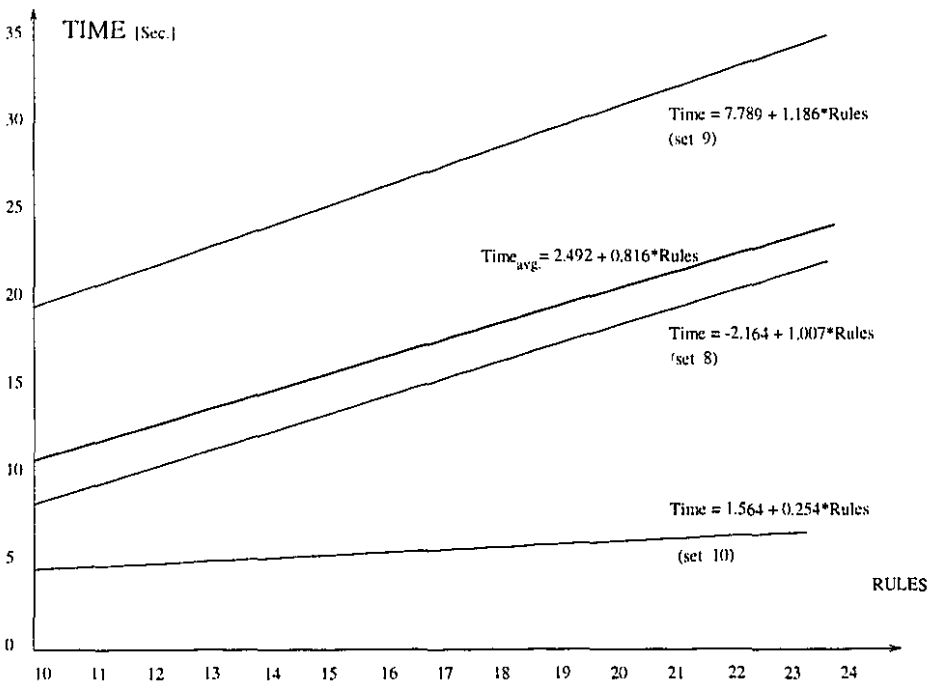

Figure 5.3: The relationship between TIME and RULES for three beds.

and the average equation for 3 beds becomes:

$$TIME_{avg} = 2.492 + 0.816 \times RULES \tag{5.9}$$

In table 5.4, for 1 bed we obtain:

$$TIME = 0.2794 + 0.3917 \times RULES \tag{5.10}$$

and for 7 beds:

$$TIME = 7.9697 + 2.0729 \times RULES \tag{5.11}$$

The speed of program execution $\frac{\partial rules}{\partial time} = \frac{1}{\tan\alpha} = \frac{1}{B}$ corresponding to the number of rules fired in each period of time. Thus, the speed of program execution derived from the regression analyses in figures. 5.1 to 5.3 given: as following

for 1 bed (eq. 5.10) $\tan\alpha_1 = 0.3917 \implies \angle\alpha_1 = 0.3733$

for 3 beds (eq. 5.9) $\tan\alpha_3 = 0.8156 \implies \angle\alpha_3 = 0.6842$

for 7 beds (eq. 5.11) $\tan\alpha_7 = 2.0729 \implies \angle\alpha_7 = 1.1213$

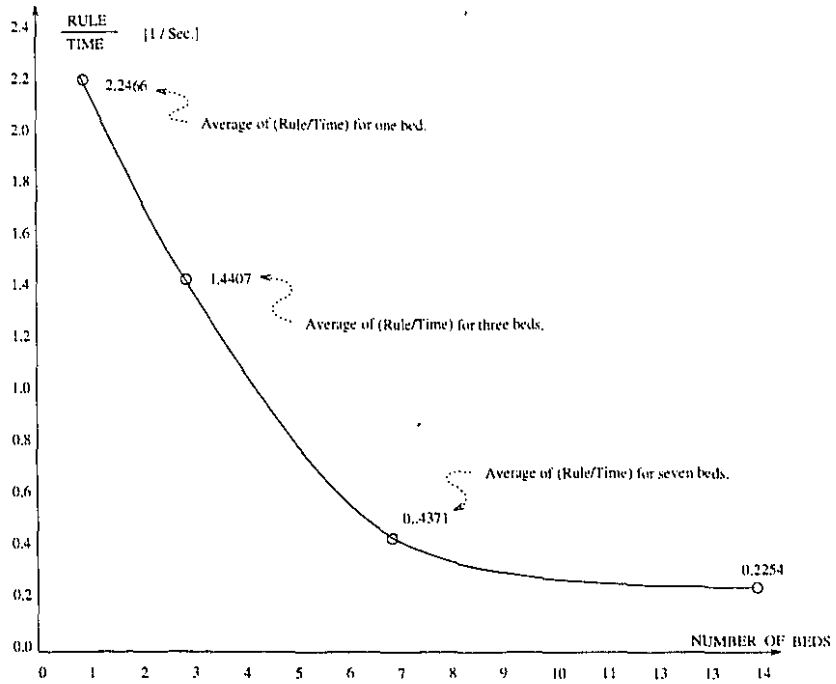

Figure 5.4: Relationship between RULES over TIME and number of beds.

Thus we see that

$$\angle\alpha_1 < \angle\alpha_3 < \angle\alpha_7 \tag{5.12}$$

When $\angle\alpha$ is very small, ∂t will be very small and/or ∂r will be very big. Executing the program with the large group of information rules decreases the speed of execution. In other words, it is better to execute the program for each bed separately and process the information concerning one bed completely, and then insert new information and run the algorithm again. The plot in Fig. 5.4 confirms this. The ratio of rules per unit time versus the number of beds (facts) is a decreasing function shown in the Fig. 5.4. The ratio of rule per unit time is maximized as the number of beds is minimized, and vice versa. Therefore , the improved CLIPS program as described in chapter 3 processes the minimum number of facts associated with a single bed. Finally, the rules which did not execute to completion with a large number of facts, were carefully studied to reveal that the design of the rules could be modified to greatly improve the program pattern matching performance. These improvements were also incorporated in the final rule design presented in chapter 3. The run time of the final version of the vital sign monitoring system to process

and store the medical condition diagnoses for fourteen beds of the intensive care unit is estimated to be less than two seconds. This processing rate will certainly meet the targeted goal of monitoring the stream of vital sign parameters being acquired every minute. Similarly, the graphics display program, can completely update the screen display with new results for fourteen beds in under two seconds, which is more than acceptable for the targeted update rate of once per minute. Further evaluation studies will be required at the ICU site to evaluate the relative merits and user acceptance of the two alternate displays. Both the expert system and the graphic display programs executed without difficulty on a diskless SUN workstation with 16MBytes of RAM attached to a networked file server. The overall execution performance of this implementation is found to be excellent.

5.4 Future Developments

It is proposed to completely migrate the patient data management system to the UNIX environment which offers a number of advantages over the OS/2 version. The TCP/IP networking capabilities under UNIX are very well established and supported. Integration with other hospital systems would thus be greatly facilitated. A challenging extension to the expert system would involve enhancing the diagnostic rules to include more comprehensive evaluation based on additional patient information such as drugs, diseases, or medical interventions. The expert system could be redesigned for implementation using Artificial Neural Systems which offer the capability of learning. A fuzzy logic methodology might also be attempted. Object Oriented programming environments such as Objective C of NeXTSTEP or C++ can greatly simplify the development and enhance portability of the system. The architecture of the Graphical User Interface as it stands cannot be extended to deal with additional information without significant modifications such as the inclusion of scroll bars, sub windows or dialog boxes which would fundamentally change the basic philosophy of the present comprehensive display.

5.5 Summary

The statistical analysis technique of lines regression was used to analyze the performance of the CLIPS program. The implementation performance was evaluated and discussed. Future research directions were suggested.

Set	Run	Fact	Rules Fired	Run Time (sec.)	Rules per Second	Fig.
1	1	1	7	2.905	2.410	5.1
1	2	1	5	2.129	2.349	5.1
1	3	1	4	1.836	2.179	5.1
1	4	1	4	1.680	2.381	5.1
2	5	1	7	3.179	2.202	5.1
2	6	1	5	2.290	2.183	5.1
2	7	1	4	1.899	2.106	5.1
2	8	1	4	1.902	2.103	5.1
2	9	1	4	1.749	2.287	5.1
2	10	1	4	2.173	1.841	5.1
3	11	1	7	2.976	2.352	5.1
3	12	1	5	2.306	2.168	5.1
3	13	1	4	1.849	2.163	5.1
3	14	1	4	1.674	2.389	5.1
4	15	14	98	434.698	0.225	-
7	16	7	47	108.402	0.434	5.2
7	17	7	33	66.360	0.497	5.2
7	18	7	27	70950	0.381	5.2
8	19	3	21	19.009	1.105	5.3
8	20	3	15	12.829	1.692	5.3
8	21	3	12	9.985	1.202	5.3
8	22	3	0	0.001	0.000	5.3
9	23	3	18	31.929	0.564	5.3
9	24	3	18	26.350	0.683	5.3
9	25	3	12	20.647	0.581	5.3
9	26	3	12	21.704	0.553	5.3
9	27	3	12	20.747	0.578	5.3
9	28	3	12	20.802	0.577	5.3
9	29	3	12	21.026	0.571	5.3
9	30	3	12	27.209	0.441	5.3
10	31	3	21	6.868	3.058	5.3
10	32	3	19	6.565	2.894	5.3
10	33	3	20	6.850	2.920	5.3
10	34	3	17	6.260	2.712	5.3
10	35	3	16	5.732	2.791	5.3
10	36	3	17	5.835	2.913	5.3
10	37	3	15	5.316	2.822	5.3
10	38	3	17	5.856	2.903	5.3

Table 5.2: Run times of the Initial CLIPS program.

Set	8	9	10
A	-2.16	7.789	1.564
B	1.007	1.186	0.254
r	0.9998	0.781	0.773
n	3	8	8
T_{avg}	13.941	23.802	6.160
$T_{\sigma(n)}$	3.767	3.9946	0.530
$T_{\sigma(n-1)}$	4.614	4.218	0.566
R_{avg}	16	13.5	18.123
$R_{\sigma(n)}$	3.742	2.598	1.615
$R_{\sigma(n-1)}$	4.583	2.777	1.727
eq. no.	5.6	5.7	5.8

Table 5.3: Statistical parameters of TIME and RULES for 3 beds.

Number of Beds	1	7
A	0.2794	7.9697
B	0.3917	2.0729
r	0.9697	0.773
n	14	3
T_{avg}	22.1819	81.9040
$T_{\sigma(n)}$	0.4828	118.8304
$T_{\sigma(n-1)}$	0.5011	23.0624
R_{avg}	4.8571	35.6667
$R_{\sigma(n)}$	1.1867	8.3799
$R_{\sigma(n-1)}$	1.2315	10.2632
eq. no.	5.10	5.11

Table 5.4: Statistical parameters of TIME and RULES for 1 and 7 beds.

Chapter 6 Conclusion

This thesis has presented a historical overview and discussed the literature associated with artificial intelligence and knowledge engineering, knowledge-based systems, rule-based systems, computer-based medical systems, hospital information systems, data base management systems, artificial intelligence in medicine, knowledge-based systems in medicine, computer based decision analysis, and interface devices for medical users. It then provided a synopsis of the hardware and software architectures of the Patient Data Management System being developed for the Intensive Care Unit of the Montreal Children's Hospital. The design of an expert system for monitoring of the ICU patient's vital signs was elaborated. The system includes a combination of C and CLIPS language programs for the automated diagnosis of patient conditions derived from on-line measurements of the heart rate, blood pressure, and central venous pressure. Two novel graphical user interfaces were developed for displaying the results of the expert system monitoring system for the ICU ward. The performance aspects were evaluated and suggested areas for future research were proposed before concluding.

Chapter 7 References

- Abel, E., Keighren, J., Rowley, D., *Distance Learning Courses in Orthopaedic and Rehabilitation Technology*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1117-1118, 1992.
- Alonso-Betanzos, A., et al., *Obstetrical Decision-Making Based on Predictive Expert Systems*, IEEE Eng. in Med. and Biol. 13th Annual Conference, Vol. 3, pp. 1300-1301, 1991.
- Alnahi, H. G., Alshawi, S., *Database Design for the use of Machine Induction to Predict Protein Secondary Structures*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 864-866, 1992.
- Barr, A. and Feigenbaum, E. A. (Eds.), *The Handbook of Artificial Intelligence*, V. II, W. Kaufmann, Inc., Los Altos, CA, 1986.
- Bernabei, A., et al., *Graphical I/O Devices for Medical Users*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 834-836, 1992.
- Blum, R. L. and Wiederhold, G., *Inferring Knowledge from Clinical Data Banks Utilizing Techniques from Artificial Intelligence*, Proceedings 3rd Annual Symposium on Comp. Applic. in Med. Care, Washinton, pp. 303-307, 1986.
- Blum, B. I., *Information Systems for Patient Care*, Springer-Verlag, New York Inc., 1987.
- Buchanan, B. G. and Shortliffe, E. H., *Rule- Based Expert Systems*, Addison-Wesley, Reading, MA, 1987.
- Catanzaite, V. A., and Greenburg, A. G., *Neurologist: A Computer Program for Diagnosis in Neurology*, Proceedings 3rd Annual Symposium on Computers in Health Care, Washington, pp. 64-71, 1986.
- Catarci, T., Ferrara, F.M., *A Knowledge-Based Architecture to Integrate Multi-Type Data in a Hospital Environment*, IEEE Eng. in Med. and Biol. 12th Annual Conference, Vol. 3, pp. 1361-1362, 1990.
- Catarci, T., et al., *On Integration of Icon and Diagrams in User Interface to Medical Information System*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 837-839, 1992.
- Celler, B.G., et al., *Medical Workstaion for Clinical Measurement*, IEEE Eng. in Med. and Biol. 12th Annual Conference, Vol. 3, pp. 1204-1205, 1990.

- Chandler, D. L., Artificial Intelligence Machines, The Ottawa Citizen, Attawa, Nov. 8, 1992.
- Chandrasekharan, B., Gomez, F., Mittal, S., and Smith, J., *An Approach To Medical Diagnosis Based on Conceptual Schemes*, Proceedings 6th IJAI, Tokyo, pp. 134-142, 1986.
- Cinquin, P., et al., *IGOR: Image Guided Operating Robot. Methodology, Applications.*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1048-1049, 1992.
- Clancey, W. J., *Model Construction Operators*, Artificial Intelligence, Vol. 53, pp. 1-115, 1992.
- D'Atri, A., et al., *Hypermedia for Interacting with Clinical Records*, IEEE Eng. in Med. and Biol. 12th Annual Conference, Vol. 3, pp. 1359-1360, 1990.
- D'Atri, A., et al., *On the Representation Management of Medical Records in a Knowledge Based System*, to appear in J. Expert Systems with Applications, 1992.
- Daalen, C. V., *Evaluating Medical Knowledge Based System*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 888-889, 1992.
- Davies, B. L., et al., *A Robotics Assistant for Prostate Sugery*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1052-1053, 1992.
- Derder, A., et al., *Toward a Formal Task Model for Biomedical Diagnosis Assistance*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 845-846, 1992.
- Dojat, M., *An Extendable Knowledge-Based System for the Control of the Mechanical Ventilation*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 920-921, 1992.
- Donnison-Speijer, J. D., *Computer in Medical Education*, IEEE Eng. in Med. and Biol. 13th Annual Conference, Vol. 3, pp. 1143-1144, 1991.
- Dresdner, D., *A Totally New Concept for Equipment Management Programs*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1126-1127, 1992.
- Elstein, A. S., Shulman, L. S., and Sprafka, S. A., *Medical Problem Solving: An Analysis of Clinical Reasoning*, Harvard University Press, Cambridge, MA., 1986.
- Entine, S. M. and Friedman, R. B., *WISAR: A MUMPS Data Base System which Utilizes Non-prime Time*, Proceedings of MUMPS Users Group Meeting, 1984.

- Fagan, L. M., Kunz, J. C., Feigenbaum, E. A., and Osborn, J. J., *Representstion of Dynamic Clinical Knowledge: Measurement Interpretation in the Intensive Care Unit*, Proceedings 6th IJCAI, Tokyo, pp. 260-262, 1984.
- Faulkner, J. R., Klein, M. H., Greist, J. H., Taves, J. E., and Bass, S., *EPIC: Information Management for Mental Health Clinicians, Administrators and Researchers*, in J. L. Crawford, S. Vitale, J. Robinson, Eds., *Computer Applications in Mental Health.* Cambridge, MA: Ballinger Publ. Co., 1988.
- Ferrer-Salvans, P., et al., *Computerized Medical Diagnosis: Wide Spectrum Problem that Can Be Integrated on Hand of the "AEDMI" Program*, IEEE Eng. in Med. and Biol. 13th Annual Conference, Vol. 3, pp. 1139-1140, 1991.
- Flemming Ulirich, *Rule-Based Systems in Computer-Aided Architectural Design*, Expert Systems for Engineering Design, pp. 93-113,Academic Press, Inc., 1988.
- Foley, van Dam, Feiner, and Hughes, *Computer Graphics - Principles and Practice*, Addison-Wesley Publishing Company, 1990.
- Fox, M. S., *The Intelligence Management System: An Overview*, Tech. Rept. CMU-RI-TR-81-4, Carnegie-Mellon University, Robotics Institute, Pittsburgh, PA, August, 1986.
- Furgeson, G. A., *Statistical Analysis in Psychology and Education*, pp. 92, McGrow Hill Book Company, Inc., New York, 1959.
- Gallant, Stephen L., *Connectionist Expert Systems*, Comm. of the ACM, 31, (2), pp. 152-169, Feb. 1989.
- Giarratano, J. and Riley, G., *Expert Systems, principles and programming*, PWS-KENT Publishing Company, 1989.
- Gilbert, R.P., et al., *Knowledge Engineering Issues in Biomedicine*, IEEE Eng. in Med. and Biol. 12th Annual Conference, Vol. 3, pp. 1285-1286, 1990.
- Giraldo, B. F., et al., *Design of an Expert System for Arrhythmia Diagnosis*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1255-1256, 1992.
- Goodall, E. V., Christiansen, C., *Patent Practice Basics for Bioengineers*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1147-1148, 1992.
- Greist, John H., and et al., *Clinical Computer Applications in Mental Health*, Journal of Medical Systems , Plenum Publishing Corp., 1987.
- Grygotis, A., *Developement of Decidion Analysis Computer Tools for Physicians*, IEEE Eng. in Med. and Biol. 13th Annual Conference, Vol. 3, pp. 1141-1142, 1991.

- Gururajan, R, Rao, B. A., et al., *An Expert System Model for Detection of Lymphocyte Subsetes*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 851-852, 1992.
- Hadzikadic, M., Elia, P., *Automated Design of Diagnostic Knowledge-Based Systems*, IEEE Eng. in Med. and Biol. 13th Annual Conference, Vol. 3, pp. 1294-1295, 1991.
- Hayes-Roth, Frederick, *Rule-Based Systems*, John Wiley, pp.21, 1987.
- Hill, J. F., *Computer Graphics*, MacMillan, 1990.
- Hirsch, B., et al., *Expert System for a Neurosurgical Clinic - Standardization and Optimization of the Interface between Clinical Information Retrieval System and Expert System*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 860-861, 1992.
- Hudson, D. L., Cohen, M. E., *The Role of Temporal Data in a Neural Network for Medical Decision Making*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1006-1007, 1992.
- Hughes, J., *Object-Oriented Databases*, Prentice Hall International U.K LTD, 1988.
- Ilomaki, A., et al., *A New Information System for the Management of Medical Equipment and the Clinical Engineering Department* IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1123-1125, 1992.
- Jager, F. et al., *Performance Measures for Algorithms to Detect Transient Ischemic ST Segment Changes*, Proc. Comp. in Cardiology, pp. 369-372, 1992.
- Jackson, P., *Introduction to Expert Systems*, MA: Addison-Wesley, 1990.
- Jiang, K. et al., *Interface Design for Bedside ICU Charting System*, IEEE Eng. in Med. and Biol. 13th Annual Conference, Vol. 3, pp. 1222-1223, 1991.
- Jordan, T. J., *The Role of Computer Based Decision Analysis in Clinical Medicine*, Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vol. 13, No. 3, 13.6-1, pp. 1099-1100, 1991.
- Kabakcioglu, A. M., et al., *Temporal Knowledge Representation / Reasoning / Learning for Medicine*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 855-857, 1992.
- Kazuhiro, I., *New Classifier of Real Time Diagnosis*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 924-925, 1992.
- Krusinska, Eva, *Computerized Assistance in Medical Decision Making*, Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vol. 13, No. 3, 13.6-2, pp. 1101-1102, 1991.

- Kurzynski, M. V., et al., *Rule-Based Medical Diagnosis with Learning: Application to the Diagnosis of Acute Renal Failure in Children*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1259-1260, 1992.
- Laguna, P., et al., *Automatic Detection of Characteristic Points in ECG*, Proc. IV International Symposium on Biomedical Engineering, Peniscola, Spain, 1991.
- Lazarro, N., et al., *Clinical Engineering and the Quality of Health Care: the Feasibility of Local Intervention*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1144-1146, 1992.
- Levy, A. H. and Baskin, A. B., *Paradigm for Medical Problem Solving*, in Proc. 12th Hawaii Int. Conf. Syst. Sci., Honolulu, Hawaii, pp.206, 1979.
- Levy, A. H. and Baskin, A. B., *Diagnosis by Computer: A Comparative Discussion of Problem Solving Methods*, in Proc. 13th Hawaii Int. Conf. Syst. Sci., Honolulu, Hawaii, 1980.
- Marin, R., et al., *Rapid Prototyping in Graphic Interface Development for Medical Expert Systems*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 840-841, 1992.
- Martelli, S., et al., *A System for Computer and Robot Asststed Knee Implantation*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1073-1074, 1992.
- Massari, P., *A SETH: A Toxicological Expert System in Adult Drugs Poisoning.*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 906-907, 1992.
- Matsuto, T., Shiina, S.I., *Laboratory Test Scheduling Based on Constraint Propagation.*, IEEE Eng. in Med. and Biol. 12th Annual Conference, Vol. 3, pp. 1334-1335, 1990.
- McCary, A. T., *Natural Language Processing for Intelligent Information Retrieval*, Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vol. 13, No. 3, 13.11-3, pp. 1160-1161, 1991.
- McDermott, J. *R1: A Rule-Based Configurer of Computer Systems*, Artificial Intelligence 19, pp.39-88, 1982.
- Montgomery, R. L., *The Role of Computer Decision Making in the Diagnosis of Orofacial Pain.*, Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vol. 13, No. 3, 13.6-5, pp. 1108-1108, 1991.
- Mora, F., et al., *An Integrated Approach to Knowledge-Based Systems for CCU Patient Management: The First Steps*, Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vol. 13, No. 3, 13.15-2, pp. 1206-1207, 1991.

- Moret-Bonillo, W., et al., *PATRICIA: An Expert System that Incorporates a Patient-Oriented Approach for the Management of ICU Patients*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 876-877, 1992.
- Mouhamed, S. et al., *Study of a Clinical Workstation for Pulmonary Disease Diagnostic*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 1282-1283, 1992.
- Newell, A. and Simon, H. A., *Human Problem Solving*, Prentice Hall, Engelwood Cliffs, N.J., 1972.
- Nii, H. P. and Aiello, N., *AGE (Attempt to Generalize): A Knowledge-Based Program for Building Knowledge Based Programs*, Proceedings of the 6th IJCAI pp. 645-655, Tokyo, 1979.
- Nykanen, P. and Saranummi, N., *Decision Support in Medicine- Where, When, How?*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 883-885, 1992.
- Orhanoudakis, S. C., et al., *A Medical Image Database for Tomographic Images*, Image Analysis and Computer Vision Lab, Inst. of Computer Science, 4th, Heraklion, Greece, 1991.
- Ovalle, A., Garbay, C., *KIDS, Distributed Expert System for Biomedical Image Interpretation*, 12th Int. Conf. on IPMI, pp. 419-433, 1991.
- Pauket, S. G., Gorry, G. A., Kassirer, J. P., and Schwartz, W. B., *Towards the Simulation of Clinical Cognition: Taking a Present Illness by Computer*, Amer. J. Med. V. 60, pp. 981-996, 1976.
- Pople, H., Myers, J., and Miller, R., *DIALOG: A Model of Diagnostic Logic for Internal Medicine*, Proceedings 4th IJCAI, Tbilisi; pp. 841, 1975.
- Pople, H., *The Formation of Composite Hypotheses in Diagnostic Problem Solving*, Proceedings 5th IJCAI, Boston; pp. 1030-1037, 1977.
- Prakash, O., et al., *Real-Time Networking of Integrared Patient Care Workstation*, IEEE Eng. in Med. and Biol. 12th Annual Conference, Vol. 3, pp. 1219-1220, 1990.
- Pressman, R., *Software Engineering a Practitioner's Approach*, McGraw Hill Book Company, Singapore, 1987.
- Rafanelli, M., et al., *A User Interface for the Intra-Operative Staging*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 962-963, 1992.
- Rau, G., et al., *Ergonomic Aspects of Knowledge-Based Systems for Clinical Monitoring Tasks*, IEEE Eng. in Med. and Biol. 12th Annual Conference, Vol. 3, pp. 1363-1364, 1990.

- Ravaux, P., Vilhelm, C., *A Neural Approach to Knowledge Base Systems.*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 928-930, 1992.
- Reggia, J. A., *A Production Rule System for Neurological Localization* Proceedings 2nd Annual Symposium on Computer Application in Medical Care, Washington, pp. 254-260, 1978.
- Reggia, J. A., *Computer-Assisted Medical Decision Making-Knowledge Based,* Proceeding 3th Annual Symposium on Computer Application in Medical Care, IEEE, 1985.
- Rialle, V., Ohayon, M., et al., *Medical Knowledge Modeling in a Symbolic-Connectionist Perspective,* Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vol. 13, No. 3, 13.6-6, pp. 1109-1110, 1991.
- Ruggiero, C., Giacomini, M., *An Expert System for Radiological Images.*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 918-919, 1992.
- Rychener, Micheal D., *Expert Systems for Engineering Design,* Academic Press, INC., London, 1988.
- Shortliffe, E. H., et al., *An Artificial Intelligence Program To Advise Physicians Regarding Antimicrobial Therapy,* Computers in Biomedical Research V. 6; pp. 544-560. 1973.
- Shortliffe, E. H., *Computer-Based Medical Consultations: MYCIN,* Elsevier, 1976.
- Shortliffe, E. H., *Knowledge-Based System in Medicine,* IEEE Eng. in Med. and Biol. 13th Annual Conference, Vol. 3, pp. 904-905, 1991.
- Sox, H. C., Blatt, M. C., et al.,*Medical Decision Making,* Boston: Butterworths, 1988.
- Staes, F., Tarantino, L., Vermier, D., *A Graphical Query Language for Medical Information Systems,* Proc. of the 13th IEEE/EMBS, 1991.
- Stanton, S., Hammer, M., and Power, B., *Reengineering: Getting Everyone on Board,* I.T. magazine, Vol. 25, No. 4, pp. 22-27, Apr. 1993.
- Starita, A., Battaglia M., Cioni, R., *NEUREX: An Expert System for the Diagnosis of Neurogenic Diseases of the Inferior Limbs.* IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 904-905, 1992.
- Stefik, M. J., *Planning with Constrains (MOLGEN: Part 1); Planning and Metaplanning (MOLOGEM: Part 2),* Artificial Intelligence 16, pp. 111-170, 1981.

- Summers, R., et al., *Casual Probabilistic Modelling for Clinical Decision Support in the High Dependency Environment*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 869-870, 1992.
- Szolovitz, P. and Pauker, S. G., *Categorical and Probabilistic Reasoning in Medical Diagnosis*, Artificial Intelligence, V. 11, pp. 115-144, 1978.
- Tarantino, L., *Towards a New Generation of Medical Information Systems: Interaction Requirements*, IEEE Engineering in Medecin and Biology 14th Annual Conference, Vol. 3, pp. 832-833, 1992.
- Uckun, S., Davant, B. M., *Qualitative Modelling as a Paradigm for Diagnosis and Prediction in Critical Care Environments*, Artificial Intelligence in Medicine, Vol. 4(2), pp. 37-54, 1992.
- VanMelle, W., *A Domain Independent Production-Rule System for Consultation Programs*, Proceeding of the 6th IJCAI; pp. 923-925, Tokyo, 1978.
- Vannobel, J. M., Toulotte J. M., *An Expert System for Aphasia Treatment using Exercises concerning Alphabetically Written Numeral Forms.*, IEEE Eng. in Med. and Biol. 14th Annual Conference, Vol. 3, pp. 922-923, 1992.
- Weiss, S., Kulikowski, C., and Safir, A., *Glaucoma Consultation by Computers*, Comp. Biol. Med. V. 8, pp. 24-40. 1978.
- Weiss, S. and Kulikowski, C., *EXPERT: A System for Developing Consultation Models*, Proceeding of the 6th IJCAI, Tokyo; pp. 942-950, 1979.
- Weiss, S., Kulikowski, C., Amarel, S., and Safir, A., *A Model-Based Method for Computer-Aided Medical Decision-Making*, Artificial Intelligence V. 11, pp. 145-172, 1983.
- Williams, B. T., *Computer Aids to Clinical Decisions*, V. 2, CRC Press, Inc., 1982.
- Windyga, P., *Knowledge Modeling for Reconstruction of Coronary Networks*, IEEE Engineering in Medecin and Biology 14th Annual Cnference, Vol. 3, pp. 947-948, 1992.
- Wood, S. L., Parvati, D., *Visualization Tools for Bioengineering Education*, IEEE Engineering in Medecin and Biology 14th Annual Cnference, Vol. 3, pp. 955-956, 1992.
- Young, M. R., et al., *Real Time Visualization of Cardiac Arrhythmias*, IEEE Engineering in Medecin and Biology 14th Annual Cnference, Vol. 3, pp. 1244-1245, 1992.