CONSTRUCTING CASE BASES FROM MEDICAL DATABASES

by

Lijuan Wang

TR97-117, December 1997

This is an unaltered version of the author's
MCS Thesis

Faculty of Computer Science
University of New Brunswick
Fredericton, N.B. E3B 5A3
Canada

Phone: (506) 453-4566
Fax: (506) 453-3566
E-mail: fcs@unb.ca
www: http://www.cs.unb.ca

CONSTRUCTING CASE BASES FROM MEDICAL DATABASES

by

Lijuan Wang

BSc (Math), Northeastern University, 1983
MSc (Stat), University of New Brunswick, 1995

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of

Master of Computer Science
in the Graduate Academic Unit of Computer Science

Supervisor:	Nickerson, B. G., BScE, MScE (UNB), PhD (RPI), CS
Co-Supervisor:	Frize, M., BASc (Ott), MPhil (Lond), DIC (Imperial), MBA (Moncton), PhD (Erasmus), EE
Examining Board:	Kurz, B. J., Dipl Ing (Stuttgart), MScE, PhD (UNB), CS
	Mcallister, A. J., BA, M.Sc(CS) (UNB), PhD (Sask), CS
External Reader:	Kaye, M., BScE (UNB), MEng (Car), EE

The thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

December 1997

ABSTRACT

A case-based reasoner that makes use of a medical domain knowledge base can be used to assist physicians in decision making. This thesis presents research on preprocessing an Intensive Care Unit (ICU) medical database to arrive at an appropriate database and case base architecture. Appropriate case bases were identified using an ICU patient model. Appropriate weights for case-based matching were after making thorough statistical analysis of the underlying data. This new approach to case-based reasoning with medical data was integrated with an existing graphical user interface IDEAS for ICUs version 2.3. A case-based reasoning shell called The Easy Reasoner was used for case-based reasoning. IDEAS for ICUs version 3.0, a user friendly clinical decision support tool, was developed using Microsoft Visual Basic 4.0. Testing was carried out using an ICU database of over 3000 patients. Testing results showed a significant improvement in the running speed compared with a previous version of case-based reasoning with the same data.

ACKNOWLEDGMENTS

The author wishes to thank her supervisor Dr. Bradford G. Nickerson for his numerous guidance and encouragement at all stages of this work. The author wishes to thank her co-supervisor, Dr. Monique Frize, for her support and financial assistance.

A special acknowledgment is extended to Dr. F. G. Solven, who generously shared his time to provide me with very valuable medical insight.

Special thanks are reserved for Kirby Ward who gave me substantial help at the beginning and throughout this research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

CDSS	Clinical Decision Support System
AI	Artificial Intelligence
ICU	Intensive Care Unit
DECH	Dr. Everett Chalmers Hospital
DLL	Dynamic Link Library
ART-IM	Automated Reasoning Tool for Information Management
CBR	Case-Based Reasoning/Reasoner
SAS	Statistical Analysis System

Chapter 1

INTRODUCTION

1.1 Computer-Based Clinical Decision Support Systems

Computer-based clinical decision support systems (CDSSs) are computer programs that make use of a medical domain knowledge base to analyze specialized problems. CDSSs are specifically designed for use by a clinician involved in patient care as a direct aid to clinical decision making. Historically, the applications of artificial intelligence (AI) and other computing and information science techniques to the field of health care have resulted in the development of CDSSs. The medical domain knowledge can be either expert knowledge from doctors and/or a patient information database. At the present time, there are many kinds of computer-based clinical decision support systems. One approach is to match characteristics of an individual patient to an existing patient data base. Patient-specific information in the form of past assessments is presented to the clinician. The computerized summary of selected characteristics of past cases may be a useful reference to assist clinicians in medical decision making. Much has been written about the theoretical and technical aspects of computer-based clinical decision support systems and their reliability, validity, and acceptability. There is some evidence suggesting that some computer-based clinical decision support systems can improve physician performance [Johnston et al, 1994].

1.2 Background

The Intensive Care Unit (ICU) is a particular site in a hospital where patients who have potentially curable critical illnesses are managed. Researchers hope to improve ICU patient outcomes and, at the same time, efficiently utilize medical resources. In the U.S., expenditure for treatment in ICUs consumes 1% of the Gross National Product (approximately $62 billion in 1992) [Esserman et al, 1995]. How much of this care is effective is not known. No one computer-based clinical decision support system is used to assist physicians in classifying the ICU patients or to show the correlation of previous patients matching a presented case. Some pilot work has been done in this area. For example, Taylor [1994] has developed a prototype system, called IDEAS for ICUs Version 2.0, that employs case-based reasoning techniques on a database of over 2000 ICU patients to assist in physician decision making. One big problem of this version of IDEAS for ICUs is its running speed is very slow. And also, the matching weights were simply chosen by the physician. IDEAS for ICUs Version 2.0 performs partial matching in the database and shows the top ten matches for the new patient case. Later, IDEAS for ICUs Versions 2.1 and 2.2 were built with changes to the way data is entered, but without case-based reasoning capabilities.

1.3 Thesis Objectives

Commonly recognized objectives of computer-based clinical decision support systems include (i) to improve patient outcomes -- as measured by mortality and other important factors (e.g., decreased length of stay in an ICU or decreased hours of ventilation), and (ii) to provide efficient utilization of scarce health care resources. In addition to that, such computer programs must be easy to use, i.e., user friendly, and have a reasonable running speed. It is not so easy to meet all of these requirements. Many factors need to be considered. For example, a patient record database contains many data types. Some data types are much more important than others for clinical decision support. This thesis presents research on (i) how to preprocess a medical database to arrive at an appropriate database and case base architecture (addressed in Chapter 5); (ii) how to determine appropriate weights for case-based matching using statistics of the underlying data; (iii) how to identify subsets of a case base using an ICU patient model; and (iv) the development of IDEAS for ICUs Version 3.0 prototype. The database used for the research is ICU93.dbf, initiated by Dr. Fred Solven at the Dr. Everett Chalmers Hospital (DECH) in Fredericton, N.B. DECH, with approximately 320 beds, is a regional hospital in the province of New Brunswick and serves approximately 160,000 people.

1.4 Literature Review

AI in medicine has a long history. As early as in 1970's, a number of successful expert systems were created. Among them, MYCIN was the first successful prototype expert system in the medical domain [Yu et al, 1979]. It can be used to assist physicians in diagnosing bacterial infections, and recommending a therapy for them. Historically, it is very important for three reasons. First, MYCIN demonstrates that AI can be used for practical real-world problems. Second, MYCIN ascertains the feasibility of the expert system shell. Third, it was the testbed of new concepts such as the explanation facility, automatic acquisition of knowledge and intelligent tutoring that are found in a number of expert systems today. MYCIN is a rule-based expert system.

Since the 1980's, researchers have been developing a new idea -- case-based reasoning. As a result, various case-based reasoners have been developed for different practical purposes. One of earlier case-based reasoners is the CASEY [Koton, 1988] system. CASEY uses case descriptions of patients with cardiovascular disease and provides a diagnosis based on similarity to other cases. It can be used to diagnose heart failure. CASEY works with a more complete model-based diagnostic system, which is the source of the initial case base of diagnoses. When a new case is considered, CASEY looks for patient cases with similar, but not necessarily identical symptoms in the case base. If CASEY finds a good match, it then tries to adapt the retrieved diagnosis, considering differences in symptoms between the old and new cases. Following the success of the CASEY system, case-based reasoning techniques are now widely used.

Two other examples of case-based reasoners in the medical domain are FLORENCE [Bradburn et al, 1993] and UKAT [El-Gamal et al, 1993]. FLORENCE is a nursing care planning system that advises on the identification of nursing diagnoses in a new client. It is based on a case-based reasoning system, but has an ancillary model-based reasoning system to advise on more complex problems. UKAT is a tool which can be used for knowledge acquisition based on acquiring medical cases and reasoning on these cases.

IDEAS for ICUs Version 2.0 was completed in 1994 [Taylor, 1994]. It was developed based on the collaboration between the University of New Brunswick and the DECH. IDEAS for ICUs Version 2.0 has been tested in a three-week pilot study at DECH.

Three more recently developed case-based reasoners in the medical domain are ROENTGEN [University of Chicago, 1996], CAMP [Case Western Reserve University, 1996] and ISIS [Kahn et al, 1997]. ISIS is designed to help primary-care physicians order the most cost-effective diagnostic imaging studies. ROENTGEN can be used to plan radiation therapy for cancer patients. CAMP plans daily menus to meet individual nutritional and personal preference requirements by retrieving and adapting menus previously designed to meet dietary and aesthetic guidelines.

Chapter 2

CASE-BASED SYSTEMS

2.1 Case-Based Reasoning

The central idea of cased-based reasoning (CBR) is that the problem solver reuses the solution from some past cases to solve a current problem. Case-based reasoning is both a paradigm for computer-based problem solvers and a model of human cognition. To most people, following an explanation of a problem's solution leads to a good understanding of the problem. Very often we do not realize, when we face a problem, that we do not construct plans to solve it from first principles. We do not reason as if we had never seen a problem like that. Rather, we try to find the best plan we have heard of or previously used that is closest to the problem at hand as an "exemplar" or a "matched case". In other words, we reason from experience. We use our own experience if we have a relevant one, or we make use of the experience of others to the extent that we can obtain information about such experience. Simply stated, case-based reasoning means reasoning from prior examples. It is a very natural reasoning process. This differs from rule-based reasoning which solves problems by chaining rules of inference together.

2.1.1. Definition

A formal definition of CBR is provided by Riesbeck et al [1989], and states "A case-based reasoner solves new problems by adapting solutions that were used to solve old problems." As a paradigm for computer-based problem solvers, this definition can be converted into four basic steps. (i) Retrieving an old case -- a problem and solution -- that resembles the new problem. Old cases reside in case memory. Case memory is a database that contains description of prior cases stored as units. Retrieving an old case involves determining what features of a problem should be considered when looking for similar cases and how to measure degrees of similarity. This step is referred to as the indexing problem. (ii) Adapting the solution of the old case to the new problem taking into account any difference between the current and previous situations. Even though the old case and new case are similar, they may not be identical. The solution of the old case, in most situations, may have to be adjusted. This step is called case adaptation. (iii) Applying the adapted solution and evaluating the results. This step is referred to as testing. (iv) Updating case memory. If the adapted solution works, a new case which consists of the problem just solved and the solution used can be formed. This new case is stored in case memory so that the new solution will be available for retrieval during future problem solving. In this way, gradually, the system becomes more and more competent as it encounters more cases. The fourth step is also considered as a part of the indexing problem.

One thing that needs to be mentioned here is that not all case-based reasoners use all four steps. Depending on the situation, some case-based reasoners may not have an adaptation step. In others, the case memory may be thought to be so well developed that it provides sufficient coverage for problems in the domain so case memory is not updated. Figure 2.1 shows a case-based reasoning flow chart which is taken from Riesbeck and Shank [1989].

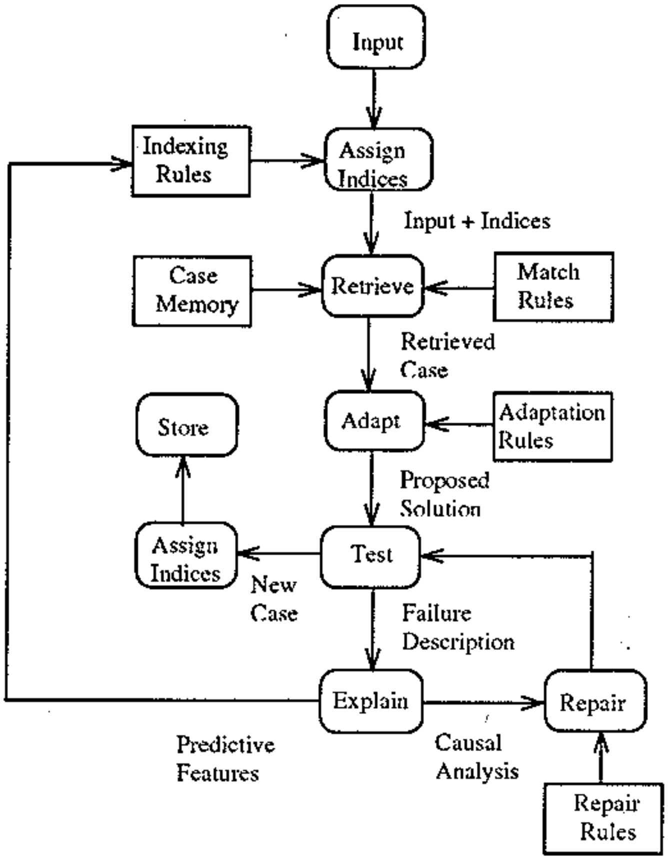


Figure 2.1: CBR flow chart (from Riesbeck and Shank [1989]).

There are two kinds of adaptation that have been described in the CBR literature. The first one is structural adaptation, in which the adaptation rules apply directly to the solution stored in a case with some modification. The second one is derivational adaptation, where the rules that generated the original solution are re-run to generate the new solution [Riesbeck and Shank, 1989].

Depending on the situation, a variety of adaptation techniques can be used in different CBR systems. They are: null adaptation, parameterized solutions, abstract and respecialization, critic-based adaptation and reinstantiation.

2.1.2. Fundamental Problems

Based on the above discussion, it is easy to understand that the basic cycle of a case-based reasoner should be "input a problem, find a relevant old solution, adapt it" [Riesbeck et al, 1989]. The case memory which stores the old cases is commonly called a case base. It is the collection of all the information of a large number of cases which have been indexed to allow case retrieval via case matching in the domain. Because a case integrates a large amount of complex information and a case base can also be very large in some domains, selecting appropriate matching cases can be a complex task. In general, the basic problems in case-based reasoning are: how to determine the most similar cases and how to identify very different ones. Which problem is more important depends on the situation. Commonly, the CBR approach in the clinical ICU setting corresponds to "show me the patients that have been here before and who are similar to this new patient".

2.2 Case-Based Reasoning Shells

Available case-based reasoning shells include CBR Express, Esteem, KATE, ReCall, MEM-1, ART-IM and The Easy Reasoner [Fenstermacher, 1995]. They are software packages that include skeletons of the main CBR functions. The users tailor the system to their own problem domain. In general, they all have their own advantages and disadvantages. Subsection 2.2.1 briefly introduces some commonly used shells -- Remind, CBR Express and ART-IM. Subsection 2.2.2 provides a detailed discussion of The Easy Reasoner, which is used for this thesis.

2.2.1 Remind, CBR Express and ART-IM

The Remind case-based reasoning tool is from Cognitive Systems. It provides decision tree-based case-based reasoning. The principal strength of Remind is its ability to induce binary decision trees from a database using a precursor to Quinlan's ID3 algorithm and to use the induced decision tree as an efficient index into the database. It supports fields that contain one or more symbols and can use the occurrence or nonoccurrence of a symbol in a field as a split criterion in its decision trees. The main problem with Remind is that its internal database has to be used, which means that data must be imported from text files to a private database format (using proprietary database software from Faircom). In general, this is an overly elaborate task.

CBR Express is a CBR tool from Inference Corporation. It provides nearest neighbor text-based case-based reasoning. The indexing of CBR Express is based on nominal and ordinal database fields with additional support for text-valued fields. Based on statistical properties of the co-occurrence of 3-1 grams (see Table 2.1), it can retrieve related cases based on text matching. This gives CBR Express better recall in the face of misspelling or morphological variations. The consequent problem is that it is very easy to present morphologically related cases that are of no semantic relevance. Another disadvantage is, just like Remind, that the internal database has to be used. Table 2.1 shows some examples of n-m gram processing for the term "POSTOP".

Table 2.1: Some examples of n-m grams.

n-m	gram
1-1	P, O, S, T, O, P
1-2	P, S, O
1-3	P, T
1-4	P, O
1-5	P, P
1-6	P
2-1	PO, OS, ST, TO, OP
2-2	PO, ST, OP
3-1	POS, OST, STO, TOP
3-2	POS, STO
4-1	POST, OSTO, STOP
4-2	POST, STOP

ART-IM is another product of Inference Corporation [Inference Corporation, 1988]. Based on the ART-IM retrieval mechanism, case-based systems can be built using the case-base reasoning utility offered by the package. IDEAS for ICUs version 2.0 is based on ART-IM [Taylor, 1994]. For different database fields, ART-IM provides special matching mechanisms according to the characteristics of the field. ART-IM can perform partial matching and also includes optional matching weight(s) and mismatching weight(s). In general, it is a good case-base reasoning tool for case matching and retrieving. The key problem with ART-IM is that its running speed is quite slow.

2.2.2 The Easy Reasoner

The Easy Reasoner is a case-based reasoning product of The Haley Enterprise. It is a case-based retrieval capability for Eclipse (see Section 2.2.2.1). In general, The Easy Reasoner is built on a high-performance production system [The Haley Enterprise, 1996]. The Easy Reasoner extends rule-based reasoning with case retrieval, thereby supporting case-based reasoning. The Easy Reasoner provides an associative memory and uses a variety of machine learning techniques, including inductive techniques, to construct decision trees. These trees can classify new information based on rules that use statistical and theoretical information from records in a database. Given a new record, its classification can be determined algorithmically by traversing the decision tree. Similar records that may have a variety of classifications can be retrieved by traversing a decision tree index constructed for a database. The Easy Reasoner can also retrieve without classification using nearest neighbor techniques. A nearest neighbor index can contain a text type field which is not considered by a decision tree index. A text type field index can be very useful when partial matching is required. In nearest neighbor retrieval, a weighted distance function (see Section 2.3.3) measures the distance between a new case and existing case stored in a database. Whether using a decision tree or nearest neighbor index, The Easy Reasoner ranks the retrieved records according to the similarity to the information provided in the query (a new case). Each non-missing value provided for a field in a retrieved record is used as a position along an axis in a N-dimensional space, where the number of dimensions is equal to the number of non-missing values.

Without using weights as scaling factors, all fields contribute evenly to the distance between a retrieved record and the point specified by a query. To coincide more closely with an expert's notion of similarity, weights can be used to refine the notion of distance. This capability enables decision-support systems to resolve problems by remembering solutions to previously encountered problems and is appropriate for tasks where performance is affected by experience or where knowledge can be acquired from existing or accumulated examples. The Easy Reasoner is, in effect, a combination of the Eclipse inference engine with the Class Induction and Indexing Engine (ClassIE) and The Intelligent Memory (TIM).

Comparing with other case-based reasoning products, The Easy Reasoner (i) goes further at the lexical level by mapping from text to symbols using English morphology and stop words; (ii) eliminates the difficulties from case-base/database integration and report generation; (iii) with Eclipse, C Application Program Interfaces (APIs) are implemented in conformance to operating system standards, such as Dynamic Link Libraries (DLLs) under Microsoft Windows and OS/2; (iv) induces its decision tree indices over standard databases. In general, the principal advantages of The Easy Reasoner versus Remind and other case-based retrieval software are: its openness, embedability, text processing and general reasoning capabilities [The Haley Enterprise, 1996]. Section 2.2.2.1 gives a brief introduction of the Eclipse inference engine; Section 2.2.2.2 gives a brief discussion about the Class Induction Engine, which is another most important component of The Easy Reasoner.

2.2.2.1 The Eclipse Inference Engine

An inference engine is a complex computer program that searches through the rules and facts in a knowledge base. An engine contains two basic components -- an inference mechanism and a control mechanism. The most common reasoning strategy used for inferencing new facts based on existing facts is modus ponens. There are two kinds of control strategy. One is backward chaining, the other is forward chaining. Backward chaining, a goal driven approach, starts with a goal or solution and works backwards in an attempt to produce facts that support that goal or solution. A forward chaining control strategy, in contrast, is data-driven. This strategy assembles facts as it works toward an unknown goal or solution. The inference engine manages the reasoning process for the system. The Eclipse inference engine is implemented in C and C++ and uses the Rete Algorithm to support both forward and backward chaining. The Rete Algorithm is the most widely known and is recognized as by far the most efficient algorithm for the implementation of production systems and the Rete Algorithm was originally developed by Carles Forgy in the course of obtaining his Ph.D. from Carnegie Mellon University in 1979 [The Haley Enterprise, 1996]. For a system with thousands of rules, non-Rete algorithm implementations tend to be much slower than those using Rete.

Eclipse is a rule-based programming language and has an embedable ANSI C library. It provides an ANSI C header file which defines the programming interface provided by the Eclipse libraries. The Eclipse API passes arguments in a manner that is compatible with most programming languages supported on popular operating systems. With Eclipse, many capabilities can be employed, including its case-based reasoning capability with The Easy Reasoner. Syntactically, Eclipse is very similar to other inference engines, including CLIPS [NASA/Johnson Space Center, 1987] and ART-IM. Conceptually, Eclipse is much more embedable than either and requires roughly one half the memory of CLIPS and one fourth the memory required by ART-IM. In terms of performance, Eclipse varies from at least two, but typically four to well over 10 times faster than CLIPS or ART-IM. These claims are supported by the NASA and Bell Northern Research Benchmark ["Eclipse Help", The Haley Enterprise, 1994]. Due to these advantages, it is anticipated that The Easy Reasoner will have a better performance and a much more reasonable running speed than the ART-IM case-based tool.

2.2.2.2 Class Induction Engine

A Class Induction Engine is used by The Easy Reasoner. This engine constructs decision trees using an induction algorithm derived from Quinlan's ID3. It is commercially supported and implemented as a C++ class library. Induction is performed over standard xBase which refers to the database and index file formats supported by dBase and SQL databases. The Easy Reasoner augments standard databases with indices that discriminate among classes rather than attributes. The engine automatically handles missing data during induction and retrieval by using probabilities [The Haley Enterprise, 1996].

The Class Induction Engine also supports nominal as well as ordinal columns, allows induction to be focused on a subset of the columns in a relational database table and automatically measures means, variances, and co-variances and supports the pruning of overly detailed decision trees.

2.3 Building the IDEAS Case Base

One of main objectives of this research is to build IDEAS case bases using ICU93.dbf. Some of the fields of the fundamental database ICU93.dbf are necessarily considered as text fields since partial matching is required. Thus, to retrieve the top ten closest matching records, the nearest neighbor index is the only type of index that can be built using The Easy Reasoner. In later sections, the word 'index' is used to represent 'nearest neighbor index'.

The Easy Reasoner capability can be incorporated into Visual Basic applications. For doing this, a modified VBXpert custom control VBXpert.vbx must be used. The original version of VBXpert.vbx is provided by Haley. The difference between the original version of VBXpert.vbx and the modified version of VBXpert.vbx is that the latter one can return a pointer to the knowledge base that recognizes the facts produced not only by Eclipse, but also by The Easy Reasoner. With VBXpert.vbx, a Visual Basic application can be a complete knowledge-based expert system that enjoys all the benefits of custom controls and the ease of modification that Visual Basic supports. Eclipse and The Easy Reasoner are implemented as a number of Dynamic Link Libraries (DLLs) that export many API functions that are available for Visual Basic application programs. All DLL functions we used to build IDEAS case bases are found in two standard modules -- eclipse.bas and cbr.bas. The eclipse.bas was provided by The Haley Enterprise and I wrote cbr.bas.

The relationship among Visual Basic, VBXpert.vbx, Eclipse and The Easy Reasoner is very subtle. Visual Basic serves as a Graphical User Interface (see Chapter 6). VBXpert.vbx is a custom control for Visual Basic. By using this custom control, a pointer to the knowledge base can be initialized. After the knowledge base is initialized, calling Eclipse and The Easy Reasoner DLL functions is possible. With the declarations in eclipse.bas and cbr.bas, many DLL functions in E2WSR.DLL, E2WSD.DLL and CBR2WS.DLL can be called to perform the functionality of Eclipse and The Easy Reasoner. Eclipse is the root inference engine and The Easy Reasoner is a case-based retrieval capability for Eclipse. The Easy Reasoner must be used with Eclipse. In some situations, depending on the requirement, to obtain a complete case-based reasoning solution, both the case-based retrieval capability of The Easy Reasoner and the rule-based reasoning capabilities of Eclipse may have to be used. Figure 2.2 shows a diagram explaining how our Visual Basic application (IDEAS for ICUs version 3.0) was built.

Sections 2.3.1 to 2.3.5 focus on discussing selected details of The Easy Reasoner such as types of fields, their treatment as ordinal or nominal values in the distance, and text field handling. These details are useful when building the IDEAS case bases.

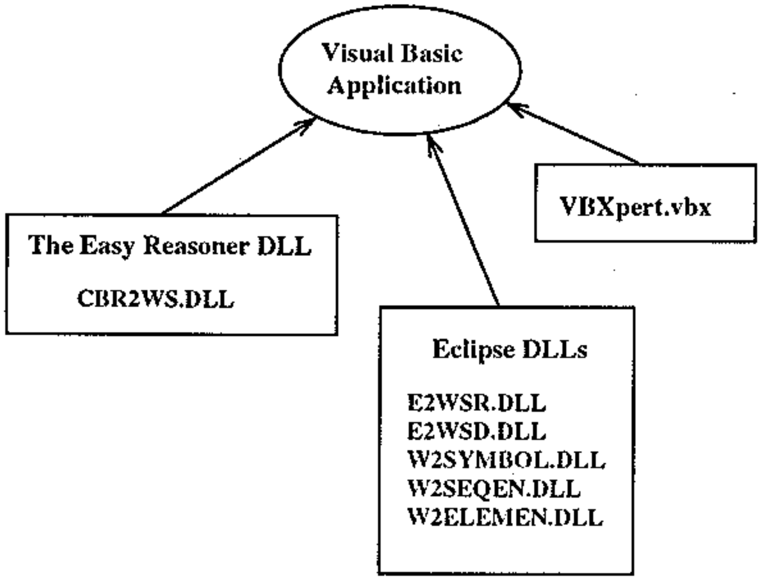


Figure 2.2: Building a Visual Basic application using Eclipse DLLs and The Easy Reasoner DLLs.

2.3.1 Building a Case Base Index

To develop a case-based reasoner, the first important problem that needs to be considered is how to build a case base index. A case base index is sometimes implemented as a pointer to a case. It represents some features of the case in the case base. In The Easy Reasoner, a case base index can be built using the information of one or more fields of an external database. In a Visual Basic application, a case base index is constructed using some specific DLL functions such as cbr_*_index(). Before these functions are called, some information of these index fields such as field name, field type and field weight must be provided by using cbr_set_*() DLL functions. The index field name must be consistent with the corresponding database field name. The index field types are boolean, data, integer, real, symbol and text. Symbol represents a contiguous string of zero or more alphabetic characters, and text is constituted by a symbol's string. Symbol and text are normal data types. The index field weight is optional and the default value is one. If the field type is text, then optional text information, which is discussed in the next section, may be provided. The index field type is a little different from the field data type defined in dBase IV. After consulting some professionals at Haley Enterprise (see Appendix V, a copy of an e-mail from Klaus P. Gross, June 25, 1997), it is now clear that both character fields and memo fields in a database file can be treated as either symbol fields or text fields in a case base index.

2.3.2 Text Processing and Term Extraction

It is said [The Haley Enterprise, 1996] that, compared to other case-based reasoning tools, one of the advantages of Easy Reasoner is its capability of retrieving based on the textual descriptions associated with records. After testing, we found that this capability works well, especially for text with long strings.

First, we would like to clarify some notions commonly used in The Easy Reasoner. In general, text processing can be seen as discarding, normalizing and extracting "terms". A "term" is defined as a sequence of contiguous alphabetic characters. This restriction, unfortunately, limits the use of The Easy Reasoner in some aspects. A symbol representing a string includes zero or more terms. A text field actually has "Eclipse symbol" as its value and it is constituted by the symbol's string. For a text field, in addition to field name, field type, and field weight discussed above, more information such as minimum word length, stop list, stemming normalization, and n-m gram extraction may be specified to control the preprocessing of the text in a symbol string. This additional information can be determined by using some DLL functions such as cbr_set_stemming() and cbr_set_nmgramming(). It is worthwhile to note that all of these operations apply at the term level.

A minimum word length defines an integer length by calling a DLL function cbr_set_min_word_length() before text processing. Any sequence of alphabetic characters which is less than this specified minimum word length in a text field is discarded.

A stop list can be defined and created by using the cbr_set_stoplist() and cbr_create_list() DLL functions. A stop list is a collection of alphabetic strings such as "of and on". In some situations, they are considered to have no contribution to matching. To accelerate matching speed, and also to reduce the errors of the final result, it would be useful to delete these strings from text before the text is passed on to term normalization or n-m gram extraction. If stop list is not specified, then each contiguous sequence of alphabetic characters is treated as a separate term, which is passed on to stemming normalization or n-m gram extraction, optionally.

For the terms in text that survive after passing though the minimum word length and the stop list criteria filters, the next processing procedure is called stem normalization. It is an algorithm to normalize English words using standard suffix conversions. An example of using stemming algorithm is that the word "closest" will be considered as "close" after it passes through the stem normalization filter. The motivation for using this algorithm is that those terms should have common suffixes removed or normalized. It is hoped that this algorithm can improve the efficiency of a case-based reasoner. Since these conversions are applied uniformly, some mistakes made in its conversion do not have a serious effect on the result. In general, it is a reliable normalization mechanism.

To effectively minimize the impact of irregular lexical variations and misspellings, the n-m gram transformation technique can be used. An n-m gram indicates that sub-sequences of alphabetic characters are to be used as terms. An n-m gram transformation dictates that the first n characters of an input term are to be an output term and that the text term should begin m characters later. This process repeats itself within an input term until there are less than n characters remaining in the input term. A limitation of this transformation occurs for a string such as "CS2635". Only the term "CS" is produced if we use a 2-1 gram. This is due to the fact that The Easy Reasoner only recognizes alphabetic ASCII characters.

If the length of an input term l is less than n, then no output term is produced. For an input term with length $l >=$ n, t = l + (l-n)/m terms will be constructed. Choosing the values of n and m can be very tricky. In general, m should be less than or equal to n and n should be less than or equal to a minimum word length, which was previously defined. However, n should not be so low that the resulting terms have little discrimination ability.

2.3.3 Similarity Assessment -- Calculating Distance

After a case base index is built, the most similar case or "atmost" (less than the total number of records) closest cases can be retrieved. The cases are retrieved in an N-dimensional space using DLL functions such as cbr_*_query and cbr_get_position_*. Each dimension of this N-dimensional space corresponds to a field in the case base index. Given a query (a new case which corresponds to a point in this N-dimensional space), a weighted distance function used by a decision tree or nearest neighbor retrieval in the N-dimensional space ranks the resulting records. The records being ranked must have non-missing values. Normalized distance (always resides between 0 and 1) between a record and the query can be obtained. The record with the minimum distance is the closest one to the query. Usually, the dimensions of an N-dimensional space must be nominal or ordinal. Ordinal fields include dates, integers, and real numbers, and nominal fields include booleans, symbols and text.

The distance between a record and the query is computed differently for ordinal fields than for nominal fields. The distance between a record and a query for an ordinal field is called an ordinal distance. The equation used to calculate the ordinal distance is given in the manual of The Easy Reasoner as [The Easy Reasoner^{tm}, pp. 30-36, 1993]

$$d_{ij} = \frac{|v_{ij} - V_j|}{|V_j^{\max} - V_j^{\min}|} \tag{1}$$

where d_{ij} represents the distance between the i-th record's value, v_{ij}, and the query's value, V_j, for the j-th field, and the maximum and minimum values for each field are determined during index construction. If all fields are ordinal fields, then the final distance d_i between a record i and a query in an N-dimensional space is defined as

$$d_i = \begin{cases} \dfrac{\sum_{j=1}^{N} d_{ij}}{N} & \text{if } \dfrac{\sum_{j=1}^{N} d_{ij}}{N} \le 1 \\ 1 & \text{if } \dfrac{\sum_{j=1}^{N} d_{ij}}{N} > 1 \end{cases} \tag{2}$$

where N is the number of ordinal fields in the case base index. When weights W_j are incorporated, equation (2) becomes

$$d_i = \begin{cases} \dfrac{\sum_{j=1}^{N} W_j d_{ij}}{\sum_{j=1}^{N} W_j} & \text{if } \dfrac{\sum_{j=1}^{N} W_j d_{ij}}{\sum_{j=1}^{N} W_j} \le 1 \\ 1 & \text{if } \dfrac{\sum_{j=1}^{N} W_j d_{ij}}{\sum_{j=1}^{N} W_j} > 1. \end{cases} \tag{3}$$

The value d_i is always greater than zero and less than or equal to one.

Equations (2) and (3) result from more than 10 experiments using two special Visual Basic test programs called test1.vbp (to test ordinal field distance) and test2.vbp (to test nominal field distance). A sample experiment can be described as follows: (i) wrote a test Visual Basic program, (ii) created a dBase IV file, (iii) constructed a case base index using the dBase IV file previously created; (iv) simulated a new case, and (v) retrieved the top 4 matched cases using some specific CBR functions (i.e. cbr_build_index(), cbr_build_query(), cbr_get_position_record_number() and cbr_get_position_distance()). The objective of doing these experiments was to find what formulae were used by The Easy Reasoner. The following example gives an illustration of an ordinal distance calculation.

Example 2.1 A test database "test1.dbf" has two fields called NUM1 and NUM2. For each field there are four records presented. This database is illustrated in Table 2.2. Given a query which is a new case (4, 18), find the top three closest matching records in the database. Here, we suppose both fields are used to construct a case base index and the weights of each field are set to 1 (default value).

Table 2.2 : Database file "test1.dbf".

Record	NUM1	NUM2
1	3	20
2	4	18
3	2	15
4	4	17

Solution:

Since the presented case is (4, 18), then according to (1), the following similarity matrix is obtained:

$$\begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \\ d_{31} & d_{32} \\ d_{41} & d_{42} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.4 \\ 0 & 0 \\ 1 & 0.6 \\ 0 & 0.2 \end{bmatrix} \qquad (4)$$

Based on (2), $d_1 = (0.5 + 0.4)/2 = 0.45$, $d_2 = (0 + 0)/2 = 0$, $d_3 = (1 + 0.6)/5 = 0.8$ and $d_4 = (0 + 0.2)/2 = 0.1$. So, the top three closest records, in descending order, are record 2, 4 and 1. Record 2 is a perfect match. This result is exactly the same as what was obtained from test1.vbp using the same data.

The distance between the value of a text field in a record and that specified in a query is noted as nominal distance. The nominal distance between two text fields at record *i* can be calculated based on the following equation:

$$s_i = 1 - \sum_{k=1}^{T} W_{ik} W_{qk}, \qquad (5)$$

where

$$W_{ik} = \frac{-F_{ik} * Log(\frac{n_k}{n} + \varepsilon)}{G_i}, \qquad (6)$$

$$W_{qk} = \frac{-F_{qk} Log(\frac{n_k}{n} + \varepsilon)}{G_q}, \qquad (7)$$

and

$$G_i = \sqrt{\sum_{k=1}^{T} (F_{ik} Log(\frac{n_k}{n} + \varepsilon))^2}, \qquad (8)$$

$$G_q = \sqrt{\sum_{k=1}^{T} (F_{qk} Log(\frac{n_k}{n} + \varepsilon))^2}. \qquad (9)$$

W_{ik} is the weight of the k-th term (appearing in the text field in the case base) in record i, and W_{qk} is the weight of the k-th term (appearing in the text field in the case base) in the query. Here, all terms appearing at the text field in the case base are supposed to be arranged in some order and no repeat arrangement is needed. The order can be defined freely but never changed after it is defined. T is the total number of terms appearing at the text field in the case base, n is the number of records in the database, n_k is the number of different records in which the k-th term occurs, n_j is the number of different records in which the j-th term occurs, F_{ik} is the number of occurrences of k-th term in record i divided by the total number of terms in record i, F_{qk} is the number of occurrences of the k-th term in the query divided by the total number of terms in the query, and ε is 0.000001. The following example gives an illustration of a nominal distance calculation.

Example 2.2 A test database file "test2.dbf" has only one field called CAT1. Four records are present for this field. The data type of CAT1 is character. The database "test2.dbf" is illustrated in Table 2.3. Given the query "what is time", we wish to find the top two closest matching records in the database. For simplicity, we will not consider stemming, stop list and n-m gramming techniques. The minimum word length here is assumed to be 0 (the default minimum word length is 4 in The Easy Reasoner).

Table 2.3 : Database file "test2.dbf".

Record	CAT1
1	time is fast
2	tim is fun
3	what time is it
4	rain is too heavy

Solution:

From Table 2.4, we know that the total number of different terms appearing at the field CAT1 in the case base is 10 and n is 4. Each term k and its associated frequency number n_k is illustrated in Table 2.4.

Table 2.4: Term illustration.

k	Term k	n_k
1	time	2
2	is	4
3	fast	1
4	tim	1
5	fun	1
6	what	1
7	it	1
8	rain	1
9	too	1
A	heavy	1

Based on the definition, for each F_{ik} in the record, it follows that:

$F_{11} = 1/3$, $F_{12} = 1/3$, $F_{13} = 1/3$, and $F_{1k} = 0$ (for k = 4, 5, 6, 7, 8, 9, A)

$F_{22} = 1/3$, $F_{24} = 1/3$, $F_{25} = 1/3$, and $F_{2k} = 0$ (for k = 1, 3, 6, 7, 8, 9, A)

$F_{31} = 1/4$, $F_{32} = 1/4$, $F_{36} = 1/4$, $F_{37} = 1/4$, and $F_{3k} = 0$ (for k = 3, 4, 5, 8, 9, A)

$F_{42} = 1/4$, $F_{48} = 1/4$, $F_{49} = 1/4$, $F_{4A} = 1/4$, and $F_{4k} = 0$ (for k = 1, 3, 4, 5, 6, 7)

and for each F_{qk} in the query "what is time", we have

$F_{q1} = 1/3$, $F_{q2} = 1/3$, $F_{q6} = 1/3$, and $F_{qk} = 0$ (for k = 3, 4, 5, 7, 8, 9, A)

The next step is to calculate the distance between record 1 and the query. The value of the ε is 0.000001. Based on equation (6), for record 1:

$W_{11} = -(F_{11}*\text{Log}(2/4+\varepsilon))/\text{sqrt}((F_{11}*\text{Log}(2/4+\varepsilon))^2+(F_{12}*\text{Log}(4/4+\varepsilon))^2+(F_{13}*\text{Log}(1/4+\varepsilon))^2)$ $= 0.100343/\text{sqrt}(0.100343^2 + 0^2 + 0.200686^2) = 0.447221$, $W_{12} = 0$, $W_{13} = 0.894442$, and $W_{1k} = 0$ for k = 4, 5, 6, 7, 8, 9, A.

Similarly, for the query "what is time",

$W_{q1} = -(F_{q1}*\text{Log }(2/4+\varepsilon))/\text{sqrt}((F_{q1}*\text{Log }(2/4+\varepsilon))^2+(F_{q2}*\text{Log}(4/4+\varepsilon))^2+(F_{q6}*\text{Log }(1/4+\varepsilon))^2)$ $= 0.100343/\text{sqrt}(0.100343^2 + 0^2 + 0.200686^2) = 0.447221$, $W_{q2} = 0$, $W_{q6} = 0.894442$, and $W_{qk} = 0$ for k = 3, 4, 5, 7, 8, 9, A.

So, according to the equation (5), $s_1 = 1 - 0.447221^2 = 1 - 0.2 = 0.8$. Similarly, $s_2 = 1$, $s_3 = 0.25$, $s_4 = 1$. So, the top two closest matching records, in descending order, are record 3 and record 1.

Symbol is a special case of text with only one term that is not subjected to minimum word length, stop list, stem normalization, or n-m gram extraction. The weights for symbol fields are computed as in the above equations (6) and (7) for text fields with F_{ik} always being either one or zero. Given these weights, the distance is computed exactly as for text fields.

It is normal that a database file contains several ordinal fields and several nominal fields. In this case, the results of the experiments show that the distance D_i between a query (a presented case) and a record i can be calculated according to the following equation:

$$D_i = \begin{cases} X_i & \mathit{if}\ X_i \le 1 \\ 1 & \mathit{if}\ X_i > 1 \end{cases}, \tag{10}$$

where

$$X_i = \frac{\sum_{j=1}^{N_1} w_j d_{ij} + \sum_{k=1}^{N_2} w_k s_{ik}}{\sum_{j=1}^{N_1} w_j + \sum_{k=1}^{N_2} w_k}, \tag{11}$$

N_1 is the number of ordinal fields, N_2 is the number of nominal fields, w_j is the field weight of an ordinal field, w_k is the field weight of a nominal field, d_{ij} is an ordinal field distance, and s_{ik} is a nominal field distance.

Chapter 3

ANALYSIS OF A MEDICAL DATABASE

3.1 Motivation

The amount of data from various sources in an ICU is quite large. These data, usually, are a very valuable reference for the physician to assist with clinical decisions. Since the amount of data is large, it is possible that some bugs or outliers exist in the data. These bugs or outliers may come from different sources such as typing mistakes, scoring system errors or simply represent a special case which needs further investigation. In hospital research, a particular method is often used to study these data without previous analysis of quality or semantic content. If bugs exist, they may affect clinical outcomes. Also, there are many clinical parameters in an ICU database and some may be highly correlated with others. This information can be obtained by doing statistical analyses. In addition to the model constructed for the ICU patient (addressed in Chapter 4), the information will assist in determining the weights for matching (addressed in Chapter 5). Considering all of the above, this chapter (i) discusses the preliminary quality of the ICU database (ICU93.dbf) in the DECH, and (ii) describes an analysis of the ICU data using carefully chosen statistical methods.

3.2 Examination of DECH/ICU Database

The ICU database in DECH (ICU93.dbf) contains 116 fields and 3245 patient records. The complete data structure and field explanation is given in Appendix I. The ICU database is intended for use as a fundamental database in the IDEAS for ICUs version 3.0. Careful examination of the database reveals that records 3228 to 3237 and records 3241 to 3245 do not have any meaningful values for each field in the ICU database. In addition to that, among 116 fields, field INFT_ICU_3 (standing for infection acquired in ICU diagnosis no. 3) does not have any value associated with it. Field OTDEATHTIM (standing for death time in other location, see Appendix I) only has two values (record 2965 -- 1 and record 3161 -- 22). Field ODEATHTIME has 2553 values with 2552 zeros and one -1. Field CHSTUBDUR3 has 2553 values with 2552 zeros and one 16. According to the discussion in the previous chapter, the fields mentioned above are not qualified to be used to build a case base index. These fields are ignored for the subsequent statistical analysis and construction of the case base index in later chapters. Since records 3228 to 3237, and records 3241 to 3245 have no meaningful values, they are removed from the DECH/ICU database.

Certain values such as '-1' for special purposes were added to some fields. This must be recognized and considered when doing data analysis. In the DECH ICU database, many numerical fields such as APACHE2 (score of Acute Physiology and Chronic Health Evaluation II prognostic system [Knaus et al, 1986]) contain '-1' as their "value". The APACHE II score system cannot produce this value. Consultation with Dr. Solven revealed that '-1' entries indicates 'data not available'. This was purposely done from the beginning of data collection to distinguish this case from actual blanks or 0 which could be real values. It is meaningless to use these '-1's when calculating statistics such as correlation coefficients. Instead, it is better to consider them as missing values when doing data analysis. They are considered when we build case bases later on.

Some extreme values exist in the DECH ICU database. For example, one of the values of DAYS_ICU is 311 days, and this may heavily detract from the analysis. However, consultation with Dr. Solven revealed that this did happen due to peculiar circumstances. Since this represents "site specific considerations", it was decided to exclude them from the data analysis. These values include APACHE2 > 70 (3 values: patient records 71, 88, 98), DAYS_ICU > 100 (2 values: patient records 2092, 3045), ARTLINDUR1 > 40 (only one value: record 2859) and U_CATH_DUR > 100 (only one value: record 2092).

Diagnoses fields are very important in the DECH ICU database. For example, AD_DX1 (admitting diagnosis no. 1) contains valuable frequency information on the different types of patients admitted into the ICU. The problem is that some values in AD_DX1 or other diagnosis fields are not consistent with the DECH/ICU CODE [DECH, 1992] which is in ICD-9 classification (see Section 3.3.1) and supposed to be part of important references when constructing the ICU database. A few mistakes also exist. For example, codes which are supposed to be POSTOP were written as POSTP or PSTOP, and the same disease "Measures" is represented using different codes MEASURES and MEASURE; the latter is not in the DECH/ICU CODE. After consulting Dr. Solven, changes have been made to improve the accuracy of the data. These changes are documented in Section 3.3.4.

3.3 Preprocessing the DECH/ICU Database

Data analysis is required on the existing ICU database to obtain a satisfactory case base structure and also to acquire match weights in different phases or time intervals (addressed in chapter 4) in the ICU. Some diagnosis fields such as AD_DX1 and CHR_DX1 (chronic diagnosis no. 1) are considered to be very important since they contain substantial ICU patient type and frequency information for the DECH in the past few years. Combining those with other important fields such as APACHE2, AGE, DAYS_ICU (days in ICU), VENTHRS (ventilation hours) and DEATH, through statistical analysis, it is anticipated that, at least, part of the information about ICU patient model construction and match weights determination could be obtained. Unfortunately, the diagnosis fields mentioned above have a character string data type, and for every field there are more than 900 possible values in the domain. To make a reasonable use of them, data preprocessing is useful.

3.3.1 ICD-9-CM International Classification of Diseases

ICD-9-CM International Classification of Diseases [HSRG, 1996] is used to do the data preprocessing. This set of codes provides a standard international classification of diseases. In ICD-9-CM International Classification of Diseases, there are 999 kinds of different diseases and each disease belongs to one of seventeen categories. According to this system, the patient diagnosis recorded in the ICU database of DECH can be divided into some reasonable groups which can be used as values of a category variable in the data analysis. We call the ICD-9-CM International Classification of Diseases as "ICD-9 Classification" in later sections.

3.3.2 Mapping and Merging

To have a text file of DECH/ICU codes, a scanner with **TypeReader Professional** software was used to scan an existing hard copy "Abbreviation for ICU Dbase (1988 through September 1992)" [DECH, 1992]. Appendix II provides a copy of the first page. A comma delimited text file "ICD_9.txt" (requiring editing on about 100 lines) was obtained. It contains all the information of DECH/ICU codes. Two C++ programs were written to do the data preprocessing. The name of the first program is ICD9_c_s.cc. This program reads the comma delimited text file "ICD9.txt" and writes the results to another external text file "ICD9_c_s.txt" which contains three columns: ICD9_code (sorted in alphabetical order), ICD9_classification (numeric value) and ICD9_class (integer 0-18). Depending upon the requirement, an optional ICD9_explanation column (character string) can be written into the output file. The name of the second program is ICUClass.cc. This program reads the comma delimited text file "ICUComma.txt" which is the ICU database (ICU93.dbf) in ASCII text format, and "ICD9_c_s.txt". The output file after running this program is a classified file "ICUClass.txt" whose contents depend upon the use of the statistical analysis. A small modification is needed to obtain different "ICUClass.txt" files. The following diagram (Figure 3.1) shows the preprocessing process.

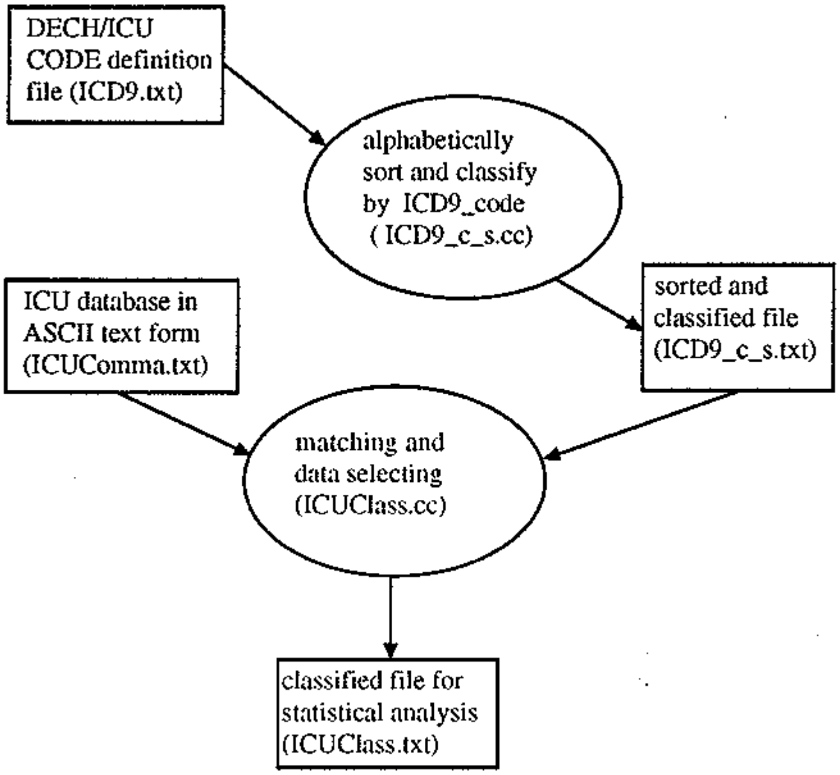


Figure 3.1: ICU database preprocessing.

3.3.3 Format of Files

(1). Figure 3.2 shows the first 4 lines of file ICD9.txt. The entire file has 759 lines, 27476 bytes, one line per diagnosis code.

```
"Abdominal Abscess","ABSC-ABD","567.2",
"Abd. Aortic Aneurysm","AAA","441.4",
"AAA Leak","LEAK-AAA",
"AAA Repair","REP-AAA","39.52",
```

Figure 3.2: First four lines of ICD9.txt.

(2). Figure 3.3 shows the first 4 lines of file ICD9_c_s.txt. The entire file has 759 lines, 22165 bytes, one line per diagnosis code.

```
A              0         0
AAA          441.4       7
ABD-FIST    569.81       9
ABSC         682.9      12
```

Figure 3.3: First four lines of ICD9_c_s.txt.

(3). Figure 3.4 shows the first 2 lines of file ICUComma.txt. The original database file has 3245 records. The size of ICUComma.txt is 884410 bytes. Each record in Figure 3.4 is shown as 4 lines due to its excessive length.

```
2,0,0,,xxxxxxxxxx,xxxxxxxxxx,81,M,142537,001,0,-1,12/22/87,15,OR-
4SW,0,0,,PETERS,,SURG,POSTOP,RSCT-WEDGE,PNEUMOTHOR,,,,,,,,,,,,,,,,,,,,,01/01/88,11,4SW,B,-
1,-1,-1,PETERS,SURG,-
1,0,,,0,,0,,0,,,0,,,0,,,1,1,7,0,0,1,0,0,0,0,0,,0,0,0,0,,1,1,84,0,1,3,19,8,0,0,,0,,0,,,1,0,,0,0,,,0

-1,0,,,xxxxxxxxxx,xxxxxxxxxx,37,F,136914,002,0,-
1,12/25/87,19,ER,0,0,,RANKIN,,SURG,TRAUMA,MVA,PNEUMOTHOR,,,,,,,,,,,,,,,,,,,,,01/01/88,13,4S
W,B,0,-1,-1,PETERS,SURG,-
1,,,0,0,0,0,,0,,,0,,,0,,,1,1,4,0,0,1,0,0,0,0,0,,0,0,0,0,0,,0,0,0,,1,2,5,3,0,0,,0,,0,,,1,0,,1,0,,,1,ART LINE
```

Figure 3.4: First two lines of ICUComma.txt.

(4). Figure 3.5 shows the first 2 lines of file ICUClass.txt. This file has 3245 lines, 7232 bytes, with 7 fields EPIDU_DUR, C_LINE_NUM, CCU, AD_DX1, ICD9_classification, ICD9_class and DEATH, respectively, where ",," represents a missing value.

```
2 ,0,0,POSTOP,2000,18,0
-1,0,,TRAUMA,959.9,17,0
```

Figure 3.5: First two lines of ICUClass.txt.

The main reason for having this kind of file format is to make the file easier to use in a SAS (Statistical Analysis System) program. According to ICD-9-CM Classification, the domain values of ICD9_classification (ICD-9-CM ranges) of diseases are 001 - 999. We classified ICD9_classification of diseases 001-139 to be ICD9_class (a name we gave) 1, 140-239 to be 2, 240-279 to be 3, 280-289 to be 4, 290-319 to be 5, 320-389 to be 6, 390-459 to be 7, 460-519 to be 8, 520-579 to be 9, 580-629 to be 10, 630-676 to be 11, 680-709 to be 12, 710-739 to be 13, 740-759 to be 14, 760-779 to be 15, 780-799 to be 16 and 780-999 to be 17.

There are some exceptions. For example, code POSTOP is repeated many times in the ICU database, but there is no ICD9_classification value in [HSRG, 1996] for it. To easily deal with it, we defined the "ICD9_classification" (for convenience, we still use this name) of it to be 2000 which is far away from real domain values of ICD9_classification of diseases so as to be easily recognized. The ICD9_class was defined to be 18. The number of any other diseases which do not belong to real domain values of ICD9_classification but are in the ICU database seems to be low (app. 10%); for example, "LEAK_AAA", "A" and so on. For such diseases, we simply defined the "ICD9_classification" as 0 and ICD9_class as 0 too. Once again, please note that the "ICD9_class" is a name we defined. Table 3.1 shows the detailed explanation. Further details are contained in Wang [1997].

Table 3.1: Detailed description of ICD9_class classification.

ICD9_class	ICD-9-CM ranges	Description
0	N/A	DECH codes for which no ICD-9-CM code exists except for POSTOP
1	001-139	Infections and parasitic Diseases
2	140-239	Neoplasms
3	240-279	Endocrine, Nutritional, and Metabolic Diseases and Immunity Disorders
4	280-289	Diseases of the Blood and Blood-forming Organs
5	290-319	Mental Disorders
6	320-389	Diseases of the Nervous System and Sense Organs
7	390-459	Diseases of the Circulatory System
8	460-519	Diseases of the Respiratory System
9	520-579	Diseases of the Digestive System
10	580-629	Diseases of the Genitourinary System
11	630-679	Diseases of the Pregnancy, Child Birth, and the Puerperium
12	680-709	Diseases of the Skin and Subcutaneous Tissue
13	710-739	Diseases of the Musculoskeletal System and Connective Tissue
14	740-759	Congenital Abnormalities
15	760-779	Certain Conditions Originating in the Perinatal Period
16	780-799	Symptoms, Signs and Ill-defined Conditions
17	800-999	Injury and Poisoning
18	N/A	POSTOP

3.3.4 Necessary Diagnostic Code Changes

During the process of data preprocessing, it was found that for some diseases, name codes in ICD9.txt are defined slightly differently from ICUComma.txt. For consistency, some codes in ICD9.txt have been changed according to the codes of the same disease names in ICUComma.txt. We did this mainly for the following reasons: (i) ICD9.txt is easier to modify than ICUComma.txt; (ii) Codes in ICUComma.txt are believed to be commonly used in DECH by the physician if they are different from the ones in ICD9.txt for the same disease; (iii) The most important consideration is that everything must be consistent. Figure 3.6 shows the changes in detail.

Codes in ICD9.txt	Codes in ICUComma.txt
ANAST.BRK	ANAST-BRK
*CA-BLADDER	CA-BLADDER
Shock-CG	SHOCK-CG
CBD EXPLOR	CBD-EXPLOR
ESOPH.VAR	ESOPH-VAR
ESOPH.REFL	ESOPH-REFL
ILIAC AN.	ILIAC-AN
INF.MONO	INF-MONO
HEP DIS	HEP-DIS
HEP METS	HEP-METS
LOB.COLLAP	LOB-COLLAP
MES.THROM.	MES-THROM
NEAR.DROWN	NEAR-DROWN
NEUR.BLADD	NEUR-BLADD
PERF-DIV.	PERF-DIV
PER.NEUROP	PER-NEUROP
PROG.PARAL	PROG-PARAL
RAD.NECK	RAD-NECK
SPIN.STEM.	SPIN-STEM
UNKNOW 1	UNKNOW-1
VALVE DIS.	VALVE-DIS
V.ARRHYTHM	V-ARRHYTHM
WND.CLOSUR	WND-CLOSUR
WND.DEHISC	WND-DEHISC

Figure 3.6: Codes in ICD9.txt were changed to match those that in ICUComma.txt.

In ICUComma.txt, two different codes were given for one kind of disease in some cases. For example, both SHOCK and SHOCK- appear in the ICUComma.txt file. In this case, we tried to build consistency by making some very slight but necessary changes to ICUComma.txt according to ICD9.txt. Otherwise, we picked the one having the most repetitions. Figure 3.7 shows these changes. The changes were also made in ICU93.dbf, ICU93_P.dbf and ICU93_O.dbf (see Chapter 5 for their definitions).

HERNIA REP	to	HERNIA-REP
DRUG RXN	to	DRUG-RXN
MI.ACUTE	to	MI-ACUTE
PUL-EMBOL	to	PUL.EMBOL
P.ROCK	to	PROCK
S.ASTHAM	to	S.ASTHAM
E.COLI	to	E.COLI
DRUG O.D	to	DRUGOD
DRUG-OD	to	DRUGOD
DRUG-OOD	to	DRUGOD
PANCITIS	to	PANCITIS-
PANSITIS	to	PANCITIS-
PANCITIS	to	PANCITIS-
POST-OP	to	POSTOP
POSTP	to	POSTOP
PSTOP	to	POSTOP
POSOTP	to	POSTOP
PBSTOP	to	POSTOP
SHCOK-SEPT	to	SHOCK-SEPT
AIROBS	to	AIROBS-
ANAPHLAX	to	ANAPHYLAX
SHOCK-	to	SHOCK
GIBLEED	to	GIBLEED-
RF-HYOXIC	to	RF-HYPOXIC
SEIZURE	to	SEIZURES
TAUMA	to	TRAUMA

Figure 3.7: Code changes in ICUComma.txt

3.4 Statistical Analysis of DECH/ICU Database

To do the statistical analysis for the DECH/ICU database, we used a revised version ICUClass.txt, defined in section 3.3.2, as the basic raw data file. File ICUClass.txt contains all the fundamental information of the DECH/ICU database. In ICUClass.txt there are 3230 patient subjects and for each patient subject there are some observations which correspond to 114 random variables. Among 114 variables, 112 variables are from the ICU database (4 fields were taken away according to section 3.2). Among the others, one is ICD9_class which was obtained based on admission diagnosis 1, and one is postop - a binary variable which indicates if a patient was from POSTOP or not. The domain values of ICD9_class are 0 to 18 which were defined as in section 3.3.3. The domain values of postop are 1 and 0, where '1' represents 'in POSTOP group' and '0' represents 'not in

POSTOP group'. Depending on the situation, different variables may play different roles in ICU patient management. However, compared with other variables, AD_DX1, APACHE2, DAYS_ICU, VENTHRS, and DEATH are definitely more important ones for our research. We have investigated the relationship of these important variables and others and also investigated the relationship among them.

After doing simple frequency counts, it was found that there are approximately two hundred kinds of DECH/ICU codes appearing in the AD_DX1 (admission diagnosis no. 1). Among them, POSTOP is repeated 1712 times which is over one-half of the observations of AD_DX1. If we divide the whole ICU data set into two groups, POSTOP and NON-POSTOP, which can be done by using the variable postop, does a relationship that we found for the whole ICU data set still hold true for both the POSTOP group and the NON-POSTOP group? Are the highly correlated variables the same in both groups?

The situation is even more complicated. Based on the domain values of variable ICD9_class, in theory, we have 19 classes. More questions can be asked: how many patient records are in each class? Is the number of observations of a variable in a ICD9_class sufficient to obtain significant statistical results? Do the highly correlated variables we mentioned above still play the same role in a special ICD9_class? I will present some simple statistics but, in this thesis, will not explore the relationship among the different classes. That is considered as interesting future work.

To do the data analysis, SAS (Statistical Analysis System) for windows Version 6.01 was used. A SAS program name class.sas was written (See Appendix III). The raw data file used to produce SAS data sets is ICUClass.txt. The SAS data sets were created by using INFILE and other data modified commands. The main SAS procedures used are CHART, CORR, UNIVARIATE, FREQ, GLM and others with some specific options. After running class.sas, some significant results were obtained, described below.

3.4.1 Methodology of Statistical Analysis

In the SAS data sets that were created, based on the DECH/ICU database, there were two kinds of variables: quantitative and qualitative variables. Quantitative variables describe attributes in terms of numerical measurement such as AGE, APACHE2, and DAYS_ICU. Qualitative variables describe attributes that are inherently non-numeric or that have been measured according to standards that do not lend themselves to numerical expression. Qualitative variables can be divided into categorical or nominal variables such as SEX and AD_DX1 and ordinal variables (please note that this ordinal variable is not related to the ordinal distance defined in Chapter 2). For quantitative variables which are normally distributed, the Pearson correlation method can be used to test the relationship among them. For quantitative variables which are not guaranteed to be normally distributed, and ordinal variables, the Spearman rank correlation method can be employed. The Spearman rank correlation method is a non-parametric method. For nominal variables, directly calculating any kind of correlation coefficients is meaningless. To do the statistical analysis, in general, the chi-square procedure is commonly used [Schefler, 1984].

Because each variable can be reasonably viewed as a random variable which has independent observations and every two variables can be viewed as paired-data, correlation analysis will be a suitable method to use whenever it is applicable. After carefully examining the normal plots of each numerical variables, it has been found that the normal distribution assumption, or even approximately normal distribution is not very suitable for all of them. Under this condition, for simplicity, the Spearman correlation method was used whenever it was applicable.

Most qualitative variables such as SEX, ARTLIN_COM (standing for arterial line catheter complications or not) in our data set cannot be considered as ordinal data. Since their sample size is large enough and we only have a few kinds of values in each variable, we decided to use the chi-square method to test the general relationship between them.

For the relationship between a categorical variable and a numerical variable, analysis of variance is probably the best method to analyze the data [Mood et al, 1974].

3.4.2 Conclusions

It has been found that for AD_DX1 the disease with the highest frequency in DECH/ICU database is POSTOP which is associated with 1712 patient records (53% of all patient records) and successive diseases with more than 100 records (see Figure 3.8) are RF-AHC (187 records), DRUGOD (152 records) and TRAUMA (117 records).

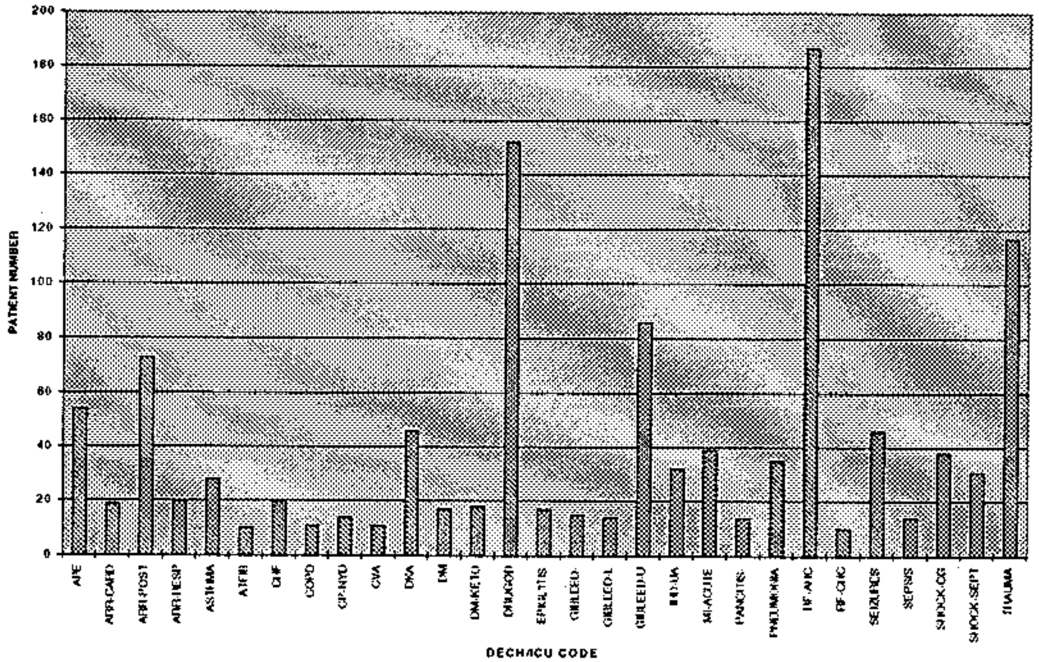


Figure 3.8: Histogram of patient number frequency counts for AD_DX1 (patient number ≥10 in NON-POSTOP group).

No patient data was in ICD9_class 11 (Diseases of Pregnancy, Child Birth, and the Puerperium), ICD9_class 13 (Diseases of the Musculoskeletal System and Connective Tissue), ICD9_class 14 (Congenital Abnormalities) or ICD9_class 15 (Certain Conditions Originating in the Perinatal Period).

It has been found that except for ICD9_class 18 (POSTOP -- patient after surgery) and ICD9_class 0 (Others), the following classes have the most patient records (over 200 patient records): ICD9_class 8 (370 patient records) which represents "Diseases of the Respiratory System", ICD9_class 17 (299 patient records) which represents "Symptoms, Signs and I11-defined Conditions" and ICD9_class 7 (232 patient records) which represents "Diseases of the Circulatory System" (see Figure 3.9).

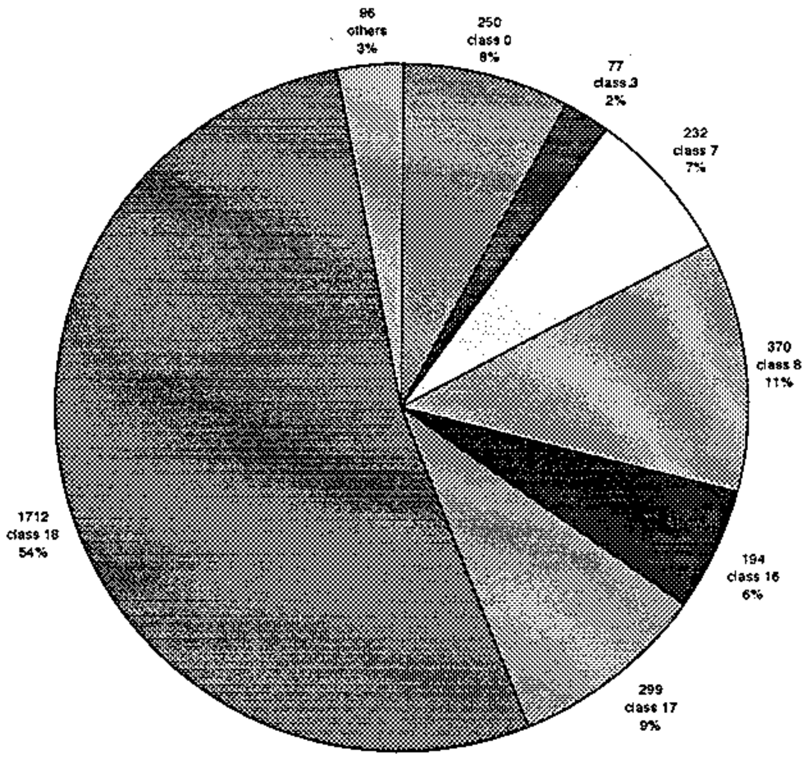


Figure 3.9 : Pie chart of patient number for AD_DX1 grouped by ICD9_class.

It has been found that DAYS_ICU has the highest Spearman correlation coefficient (0.83) with VENTHRS. The other highly correlated variables with DAYS_ICU, in decreasing order, are VENT_NUM (0.78), C_LINEDUR1 (0.58), PACATHDUR1 (0.56) and C_LINE_NUM (0.55) for the entire ICU data set (see Table 3.2).

Table 3.2: Spearman correlation coefficients for all numerical variables with DAYS_ICU (whole ICU data set).

Variable name	Spearman correlation coefficient	P-value	Number of observations
EPIDU_DUR	0.01849	0.3371	2698
C_LINE_NUM	0.55367	0.0001	2773
AGE	0.14092	0.0001	3161
APACHE2	0.53505	0.0001	1581
DAYS_CNC	-0.32946	0.0001	3147
DAYS_FLR	-0.15748	0.0001	3147
DIS_DELAY	-0.01065	0.5684	2868
IDEATHTIME	0.28503	0.0001	2572
FLDEATTIM	0.06921	0.0005	2515
ARTLIN_NUM	0.4316	0.0001	2947
ARTLINDUR1	0.51079	0.0001	2947
ARTLINDUR2	0.36272	0.0001	2586
ARTLINDUR3	0.1801	0.0001	2558
PACATH_NUM	0.54729	0.0001	2666
PACATHDUR1	0.55733	0.0001	2664
PACATHDUR2	0.21827	0.0001	2586
PACATHDUR3	0.11618	0.0001	2567
C_LINEDUR1	0.58329	0.0001	2768
C_LINEDUR2	0.32109	0.0001	2548
C_LINEDUR3	0.22185	0.0001	2549
VENT_NUM	0.78196	0.0001	2776
VENTHRS	0.83001	0.0001	2777
CHSTUBNUM	-0.0197	0.3126	2629
CHSTUDUR1	-0.00749	0.7011	2627
CHSTUDUR2	0.00019	0.9924	2590
U_CATHDUR	0.53533	0.0001	3010

It has been found that VENTHRS has the highest Spearman correlation coefficient (0.96) with VENT_NUM. The other highly correlated variables, in decreasing order, are DAYS_ICU (0.83), C_LINEDUR1 (0.61), ARTLINDUR1 (0.60), and U_CATH_DUR (0.59) for the whole ICU data set (see Table 3.3).

Table 3.3: Spearman correlation coefficients for all numerical variables with VENTHRS (whole ICU data set).

Variable name	**Spearman correlation coefficient**	**P-value**	**Number of observations**
EPIDU_DUR	0.13039	0.0001	2591
C_LINE_NUM	0.58721	0.0001	2704
AGE	0.15282	0.0001	2784
APACHE2	0.53346	0.0001	1209
DAYS_ICU	0.83001	0.0001	2777
DAYS_CNC	-0.23113	0.0001	2767
DAYS_FLR	-0.11162	0.0001	2768
DIS_DELAY	-0.00883	0.6516	2618
IDEATHTIME	0.30023	0.0001	2567
FLDEATHTIM	0.08864	0.0001	2509
ARTLIN_NUM	0.5454	0.0001	2745
ARTLINDUR1	0.60341	0.0001	2743
ARTLINDUR2	0.40219	0.0001	2582
ARTLINDUR3	0.19475	0.0001	2556
PACATH_NUM	0.53837	0.0001	2645
PACATHDUR1	0.55132	0.0001	2642
PACATHDUR2	0.24386	0.0001	2581
PACATHDUR3	0.15485	0.0001	2565
C_LINEDUR1	0.61213	0.0001	2699
C_LINEDUR2	0.33299	0.0001	2549
C_LINEDUR3	0.234	0.0001	2550
VENT_NUM	0.96123	0.0001	2782
CHSTUBNUM	0.07197	0.0003	2581
CHSTUDUR1	0.0836	0.0001	2579
CHSTUDUR2	0.04984	0.0117	2559
U_CATH_DUR	0.59345	0.0001	2742

It has been found that DAYS_ICU has the highest Spearman correlation coefficient (0.85) with VENTHRS. The other highly correlated variables, in decreasing order, are VENT_NUM (0.80), PACATH_DUR1 (0.70), PACATH_NUM (0.68) and C_LINEDUR1 (0.66) for the POSTOP group (see Table 3.4).

Table 3.4: Spearman correlation coefficients for all numerical variables with DAYS_ICU (in POSTOP group).

Variable name	Spearman correlation coefficient	P-value	Number of observations
EPIDU_DUR	0.07958	0.0023	1462
C_LINE_NUM	0.6166	0.0001	1454
AGE	0.11635	0.0001	1664
APACHE2	0.534	0.0001	874
DAYS_CNC	-0.32933	0.0001	1655
DAYS_FLR	-0.14005	0.0001	1656
DIS_DELAY	0.05635	0.0282	1516
IDEATHTIME	0.24147	0.0001	1329
FLDEATHTIM	0.0623	0.0239	1314
ARTLIN_NUM	0.37533	0.0001	1580
ARTLINDUR1	0.50746	0.0001	1580
ARTLINDUR2	0.34855	0.0001	1342
ARTLINDUR3	0.15831	0.0001	1329
PACATH_NUM	0.67897	0.0001	1407
PACATHDUR1	0.69639	0.0001	1405
PACATHDUR2	0.2147	0.0001	1339
PACATHDUR3	0.14285	0.0001	1333
C_LINEDUR1	0.66301	0.0001	1450
C_LINEDUR2	0.27786	0.0001	1323
C_LINEDUR3	0.199	0.0001	1323
VENT_NUM	0.7954	0.0001	1435
VENTHRS	0.84589	0.0001	1436
CHSTUBNUM	-0.06057	0.0242	1385
CHSTUDUR1	-0.03643	0.1757	1383
CHSTUDUR2	-0.0206	0.4484	1356
U_CATH_DUR	0.54805	0.0001	1590

It has been found that VENTHRS has the highest Spearman correlation coefficient (0.96) with VENT_NUM. The other highly correlated variables, in decreasing order, are DAYS_ICU (0.85), PACATH_DUR1 (0.65), C_LINEDUR1 (0.65) and PACATH_NUM (0.63) for the POSTOP group (see Table 3.5).

Table 3.5: Spearman correlation coefficients for all numerical variables with VENTHRS (in POSTOP group).

Variable name	**Spearman correlation coefficient**	**P-value**	**Number of observations**
--------	-----------	--------	-------------
EPIDU_DUR	0.16039	0.0001	1363
C_LINE_NUM	0.61127	0.0001	1413
AGE	0.09928	0.0002	1440
APACHE2	0.57314	0.0001	650
DAYS_ICU	0.84589	0.0001	1436
DAYS_CNC	-0.27363	0.0001	1429
DAYS_FLR	-0.10496	0.0001	1431
DIS_DELAY	0.05912	0.0291	1363
IDEATHTIME	0.24471	0.0001	1330
FLDEATHTIM	0.0578	0.0363	1313
ARTLIN_NUM	0.44751	0.0001	1424
ARTLINDUR1	0.55512	0.0001	1423
ARTLINDUR2	0.35539	0.0001	1340
ARTLINDUR3	0.15381	0.0001	1328
PACATH_NUM	0.63394	0.0001	1391
PACATHDUR1	0.65418	0.0001	1388
PACATHDUR2	0.23455	0.0001	1338
PACATHDUR3	0.1661	0.0001	1333
C_LINEDUR1	0.64707	0.0001	1408
C_LINEDUR2	0.29872	0.0001	1325
C_LINEDUR3	0.19229	0.0001	1325
VENT_NUM	0.95958	0.0001	1439
CHSTUBNUM	0.00958	0.7259	1342
CHSTUDUR1	0.03195	0.2426	1340
CHSTUDUR2	0.01327	0.6288	1330
U_CATH_DUR	0.54062	0.0001	1417

It has been found that APACHE2 has the highest Spearman correlation coefficient (0.57) with VENTHRS. The other highly correlated variables, in decreasing order, are VENT_NUM (0.56), DAYS_ICU (0.53), PACATHDUR1 (0.44), and CHSTUDUR1 (0.43) for the POSTOP group (see Table 3.6).

Table 3.6: Spearman correlation coefficients for all numerical variables with APACHE2 (in POSTOP group).

Variable name	Spearman correlation coefficient	P-value	Number of observations
EPIDU_DUR	0.12863	0.0005	734
C_LINE_NUM	0.40685	0.0001	673
AGE	0.33776	0.0001	920
DAYS_ICU	0.53400	0.0001	874
DAYS_CNC	-0.08796	0.0078	914
DAYS_FLR	-0.08427	0.0109	913
DIS_DELAY	0.04241	0.2402	769
IDEATHTIME	0.25147	0.0001	545
FLDEATHTIM	0.14441	0.0007	546
ARTLIN_NUM	0.27475	0.0001	823
ARTLINDUR1	0.37210	0.0001	825
ARTLINDUR2	0.23031	0.0001	557
ARTLINDUR3	0.09129	0.0333	544
PACATH_NUM	0.43225	0.0001	619
PACATHDUR1	0.43552	0.0001	619
PACATHDUR2	0.20563	0.0001	554
PACATHDUR3	0.15561	0.0002	550
C_LINEDUR1	0.43270	0.0001	671
C_LINEDUR2	0.22855	0.0001	538
C_LINEDUR3	0.11674	0.0067	538
VENT_NUM	0.56575	0.0001	651
VENTHRS	0.57314	0.0001	650
CHSTUBNUM	-0.03209	0.4311	604
CHSTUDUR1	-0.02147	0.5984	604
CHSTUDUR2	0.0690	0.8685	578
U_CATH_DUR	0.38564	0.0001	856

It has been found that AGE has the highest Spearman correlation coefficient (0.34) with APACHE2. The other highly correlated variables, in decreasing order, are CHSTUDUR1 (-0.21), CHSTUBNUM (-0.20), PACATHDUR1 (0.17) and PACATH_NUM (0.16) for the POSTOP group (see Table 3.7).

Table 3.7: Spearman correlation coefficients for all numerical variables with AGE (in POSTOP group).

Variable name	Spearman correlation coefficient	P-value	Number of observations
EPIDU_DUR	-0.09781	0.0002	1487
C_LINE_NUM	0.12100	0.0001	1463
APACHE2	0.33776	0.0001	920
DAYS_ICU	0.11635	0.0001	1664
DAYS_CNC	-0.08013	0.0009	1701
DAYS_FLR	0.01438	0.5531	1700
DIS_DELAY	0.05710	0.0255	1531
IDEATHTIME	0.09525	0.0005	1331
FLDEATHTIM	0.14714	0.0007	1319
ARTLIN_NUM	0.08000	0.0013	1611
ARTLINDUR1	0.04356	0.0805	1611
ARTLINDUR2	0.08720	0.0014	1344
ARTLINDUR3	0.02057	0.4533	1331
PACATH_NUM	0.15926	0.0001	1409
PACATHDUR1	0.16908	0.0001	1407
PACATHDUR2	0.04157	0.1281	1341
PACATHDUR3	0.03687	0.1780	1336
C_LINEDUR1	0.13481	0.0001	1548
C_LINEDUR2	0.02311	0.4006	1325
C_LINEDUR3	-0.00261	0.9242	1325
VENT_NUM	0.08439	0.0014	1440
VENTHRS	0.09928	0.0002	1440
CHSTUBNUM	-0.20381	0.0001	1391
CHSTUDUR1	-0.21123	0.0001	1389
CHSTUDUR2	-0.06207	0.0219	1363
U_CATH_DUR	0.05467	0.0275	1627

It has been found that ARTLINDUR1 has the highest Spearman correlation coefficient (0.78) ARTLIN_NUM. The other highly correlated variables, in decreasing order, are U_CATH_DUR (0.57), VENTHRS (0.56), DAYS_ICU (0.51) and VENT_NUM (0.50) for the POSTOP group (see Table 3.8).

Table 3.8: Spearman correlation coefficients for all numerical variables with ARTLINDUR1 (in POSTOP group).

Variable name	Spearman correlation coefficient	P-value	Number of observations
EPIDU_DUR	0.36818	0.0001	1454
C_LINE_NUM	0.39847	0.0001	1442
AGE	0.04356	0.0805	1611
APACHE2	0.37210	0.0001	825
DAYS_ICU	0.50746	0.0001	1580
DAYS_CNC	0.10140	0.0001	1600
DAYS_FLR	0.06922	0.0056	1600
DIS_DELAY	0.08141	0.0018	1471
IDEATHTIME	0.09321	0.0007	1326
FLDEATHTIM	-0.00089	0.9743	1311
ARTLIN_NUM	0.78149	0.0001	1606
ARTLINDUR2	0.24470	0.0001	1340
ARTLINDUR3	0.09463	0.0006	1327
PACATH_NUM	0.43841	0.0001	1401
PACATHDUR1	0.45486	0.0001	1402
PACATHDUR2	0.17894	0.0001	1337
PACATHDUR3	0.12781	0.0000	1332
C_LINEDUR1	0.45600	0.0001	1440
C_LINEDUR2	0.19336	0.0001	1321
C_LINEDUR3	0.12999	0.0001	1321
VENT_NUM	0.50250	0.0001	1442
VENTHRS	0.55512	0.0001	1423
CHSTUBNUM	0.28600	0.0001	1377
CHSTUDUR1	0.30265	0.0001	1375
CHSTUDUR2	0.25294	0.0001	1354
U_CATH_DUR	0.57424	0.0001	1560

It has been found that DAYS_ICU has the highest Spearman correlation coefficient (0.81) with VENTHRS. The other highly correlated variables, in decreasing order, are VENT_NUM (0.78), ARTLINDUR1 (0.61), ARTLIN_NUM (0.58) and U_CATH_DUR (0.56) for the NON-POSTOP group (see Table 3.9).

Table 3.9: Spearman correlation coefficients for all numerical variables with DAYS_ICU (in NON-POSTOP group).

Variable name	Spearman correlation coefficient	P-value	Number of observations
EPIDU_DUR	0.02099	0.461	1236
C_LINE_NUM	0.52215	0.0001	1319
AGE	0.18887	0.0001	1497
APACHE2	0.51996	0.0001	707
DAYS_CNC	-0.32562	0.0001	1492
DAYS_FLR	-0.15246	0.0001	1491
DIS_DELAY	-0.08104	0.0029	1352
IDEATHTIME	0.31474	0.0001	1243
FLDEATHTIM	0.07454	0.0098	1201
ARTLIN_NUM	0.58299	0.0001	1367
ARTLINDUR1	0.60906	0.0001	1367
ARTLINDUR2	0.37785	0.0001	1244
ARTLINDUR3	0.20026	0.0001	1229
PACATH_NUM	0.41746	0.0001	1259
PACATHDUR1	0.42118	0.0001	1259
PACATHDUR2	0.22062	0.0001	1247
PACATHDUR3	0.08633	0.0024	1234
C_LINEDUR1	0.53946	0.0001	1318
C_LINEDUR2	0.36078	0.0001	1225
C_LINEDUR3	0.2418	0.0001	1226
VENT_NUM	0.77589	0.0001	1341
VENTHRS	0.81909	0.0001	1341
CHSTUBNUM	0.11236	0.0001	1244
CHSTUDUR1	0.11351	0.0001	1244
CHSTUDUR2	0.08134	0.0042	1234
U_CATH_DUR	0.56498	0.0001	1420

It has been found that VENTHRS has the highest Spearman correlation coefficient (0.96) with VENT_NUM. The other highly correlated variables, in decreasing order, are DAYS_ICU (0.82), ARTLINDUR1 (0.70), ARTLIN_NUM (0.68) and U_CATH_DUR (0.65) for the NON-POSTOP group (see Table 3.10).

Table 3.10: Spearman correlation coefficients for all numerical variables with VENTHRS (in NON-POSTOP group).

Variable name	Spearman correlation coefficient	P-value	Number of observations
EPIDU_DUR	0.09254	0.0012	1228
C_LINE_NUM	0.57738	0.0001	1291
AGE	0.21236	0.0001	1344
APACHE2	0.47673	0.0001	559
DAYS_ICU	0.81909	0.0001	1341
DAYS_CNC	-0.19674	0.0001	1338
DAYS_FLR	-0.1196	0.0001	1337
DIS_DELAY	-0.07291	0.0098	1255
IDEATHTIME	0.36382	0.0001	1237
FLDEATHTIM	0.11698	0.0001	1196
ARTLIN_NUM	0.67809	0.0001	1321
ARTLINDUR1	0.69632	0.0001	1320
ARTLINDUR2	0.45014	0.0001	1242
ARTLINDUR3	0.23446	0.0001	1228
PACATH_NUM	0.43544	0.0001	1254
PACATHDUR1	0.44019	0.0001	1254
PACATHDUR2	0.25327	0.0001	1243
PACATHDUR3	0.14249	0.0001	1232
C_LINEDUR1	0.59237	0.0001	1291
C_LINEDUR2	0.3692	0.0001	1224
C_LINEDUR3	0.27278	0.0001	1225
VENT_NUM	0.96259	0.0001	1343
CHSTUBNUM	0.1875	0.0001	1239
CHSTUDUR1	0.18962	0.0001	1239
CHSTUDUR2	0.12368	0.0001	1229
U_CATH_DUR	0.65002	0.0001	1325

It has been found that APACHE2 has the highest Spearman correlation coefficient (0.53) with VENT_NUM. The other highly correlated variables, in decreasing order, are DAYS_ICU (0.52), VENTHRS (0.48), IDEATHTIME (0.45), AGE (0.45) and C_LINE_NUM (0.41) for the NON-POSTOP group (see Table 3.11).

Table 3.11: Spearman correlation coefficients for all numerical variables with APACHE2 (in NON-POSTOP group).

Variable name	Spearman correlation coefficient	P-value	Number of observations
EPIDU_DUR	-0.10839	0.0207	455
C_LINE_NUM	0.41264	0.0001	534
AGE	0.44742	0.0001	726
DAYS_ICU	0.51996	0.0001	707
DAYS_CNC	-0.22470	0.0001	721
DAYS_FLR	-0.14569	0.0001	718
DIS_DELAY	-0.00778	0.8469	596
IDEATHTIME	0.45195	0.0001	462
FLDEATHTIM	0.18740	0.0001	450
ARTLIN_NUM	0.37255	0.0001	587
ARTLINDUR1	0.34114	0.0001	588
ARTLINDUR2	0.18595	0.0001	460
ARTLINDUR3	0.11503	0.0153	444
PACATH_NUM	0.36724	0.0001	474
PACATHDUR1	0.37258	0.0001	474
PACATHDUR2	0.10853	0.0198	461
PACATHDUR3	0.05661	0.2318	448
C_LINEDUR1	0.41069	0.0001	553
C_LINEDUR2	0.21974	0.0001	440
C_LINEDUR3	0.18031	0.0001	440
VENT_NUM	0.52783	0.0001	558
VENTHRS	0.47673	0.0001	559
CHSTUBNUM	0.06324	0.1762	459
CHSTUDUR1	0.06435	0.1678	459
CHSTUDUR2	0.03880	0.4126	448
U_CATH_DUR	0.33091	0.0001	659

It has been found that AGE has the highest Spearman correlation coefficient (0.45) with APACHE2. The other highly correlated variables, in decreasing order, are U_CATH_DUR (0.27), C_LINE_NUM (0.22), VENTHRS (0.21) and VENT_NUM (0.21) for the NON-POSTOP group (see Table 3.12).

Table 3.12: Spearman correlation coefficients for all numerical variables with AGE (in NON-POSTOP group).

Variable name	Spearman correlation coefficient	P-value	Number of observations
EPIDU_DUR	-0.02130	0.4539	1239
C_LINE_NUM	0.20377	0.0001	1322
APACHE2	0.44742	0.0001	726
DAYS_ICU	0.18887	0.0001	1497
DAYS_CNC	-0.02128	0.4086	1501
DAYS_FLR	0.00593	0.8181	1507
DIS_DELAY	0.06465	0.0172	1357
IDEATHTIME	0.21124	0.0001	1244
FLDEATHTIM	0.11586	0.0001	1203
ARTLIN_NUM	0.16832	0.0001	1378
ARTLINDUR1	0.16536	0.0001	1378
ARTLINDUR2	0.08629	0.0023	1247
ARTLINDUR3	-0.00979	0.7316	1231
PACATH_NUM	0.16319	0.0001	1262
PACATHDUR1	0.16276	0.0001	1262
PACATHDUR2	0.06791	0.0164	1248
PACATHDUR3	0.01289	0.6509	1235
C_LINEDUR1	0.21665	0.0001	1321
C_LINEDUR2	0.05509	0.0538	1226
C_LINEDUR3	0.00633	0.8248	1227
VENT_NUM	0.20803	0.0001	1344
VENTHRS	0.21236	0.0001	1344
CHSTUBNUM	-0.05369	0.0582	1246
CHSTUDUR1	-0.05295	0.0617	1246
CHSTUDUR2	-0.02779	0.3291	1235
U_CATH_DUR	0.26695	0.0001	1435

It has been found that ARTLINDUR1 has the highest Spearman correlation coefficient (0.95) with ARTLIN_NUM. The other highly correlated variables, in decreasing order, are VENTHRS (0.70), U_CATH_DUR (0.66), VENT_NUM (0.64) and DAYS_ICU (0.61) and for the NON-POSTOP group (see Table 3.13).

Table 3.13: Spearman correlation coefficients for all numerical variables with ARTLINDUR1 (in NON-POSTOP group).

Variable name	Spearman correlation coefficient	P-value	Number of observations
EPIDU_DUR	0.18420	0.0001	1235
C_LINE_NUM	0.58166	0.0001	1304
AGE	0.16538	0.0001	1378
APACHE2	0.34114	0.0001	588
DAYS_ICU	0.60906	0.0001	1367
DAYS_CNC	0.03503	0.1945	1373
DAYS_FLR	0.01215	0.6530	1372
DIS_DELAY	-0.02039	0.4666	1277
IDEATHTIME	0.19716	0.0001	1236
FLDEATHTIM	0.10952	0.0001	1201
ARTLIN_NUM	0.95248	0.0001	1378
ARTLINDUR2	0.39867	0.0001	1247
ARTLINDUR3	0.19226	0.0006	1231
PACATH_NUM	0.41852	0.0001	1259
PACATHDUR1	0.42151	0.0001	1259
PACATHDUR2	0.23834	0.0001	1247
PACATHDUR3	0.13588	0.0001	1234
C_LINEDUR1	0.60301	0.0001	1304
C_LINEDUR2	0.35446	0.0001	1226
C_LINEDUR3	0.22485	0.0001	1227
VENT_NUM	0.64250	0.0001	1321
VENTHRS	0.69632	0.0001	1320
CHSTUBNUM	0.24116	0.0001	1243
CHSTUDUR1	0.24141	0.0001	1243
CHSTUDUR2	0.19016	0.0001	1243
U_CATH_DUR	0.66021	0.0001	1358

It has been found that the Spearman correlation coefficients for AGE and other variables (excluding APACHE2) always have relatively low values (less than 0.28) no matter which group of data is analyzed (i.e. the whole ICU data set, the POSTOP group or the NON-POSTOP group). This implies that there is no significant linear relationship between AGE and other variables such as DAYS_ICU and VENTHRS (see Table 3.7 and Table 3.12).

It has been found that some other numerical variables such as DIS_DELAY, CHSTUDUR1, and CHSTUDUR2 either have very low Spearman correlation coefficients with DAYS_ICU , VENTHRS, APACHE2, AGE and ARTLINDUR1 or have larger p-values (greater than 0.05) (see Table 3.2 - 3.13). This implies that one could not refuse the null hypothesis that there is no linear relationship between these variables and DAYS_ICU or VENTHRS with the type I error rate being 0.05.

In addition to the above discussion, it has been found that the Death Rate (DR) in POSTOP group is different from that in NON-POSTOP group (see Table 3.14)

Table 3.14 : The Death Rate (DR) in POSTOP group and NON-POSTOP group.

POSTOP	**Number of observations in death**	**DR**
NO	1509	19.8%
YES	1703	9.9%

where DR = Total Number of deaths / Total number of non-missing observations (either in POSTOP group or in NON-POSTOP group).

It has been found that for most clinical parameters, the mean value of the observations in the death group (DEATH = 1) is not the same as that in the survive group (DEATH = 0). Tables 3.15, 3.16 and 3.17 show the detailed testing results for the whole ICU data set, the POSTOP group and the NON-POSTOP group, respectively using the SNK (Student-Newman-Keuls) method.

Table 3.15 : Testing the mean for numerical variables grouped by DEATH (whole ICU data set).

Variable Name	Mean value (DEATH = 0)	Mean value (DEATH = 1)	Sign. Diff. (alpha = 0.05)
EPIDU_DUR	0.82	0.44	yes
C_LINE_NUM	0.42	0.73	yes
AGE	58.34	68.62	yes
APACHE2	12.43	20.12	yes
DAYS_ICU	1.47	3.98	yes
DAYS_CNC	1.75	1.14	yes
DAYS_FLR	1.01	0.56	yes
DIS_DELAY	0.189	0.10	yes
ARTLIN_NUM	0.61	0.75	yes
ARTLINDUR1	1.91	2.44	yes
ARTLINDUR2	0.26	0.68	yes
ARTLINDUR3	0.05	0.16	yes
PACATH_NUM	0.18	0.26	yes
PACATHDUR1	0.52	0.74	yes
PACATHDUR2	0.09	0.17	no
PACATHDUR3	0.03	0.09	yes
C_LINEDUR1	1.44	2.28	yes
C_LINEDUR2	0.18	0.81	yes
C_LINEDUR3	0.12	0.56	yes
VENT_NUM	0.35	0.71	yes
VENTHRS	32.38	74.16	yes
CHSTUBNUM	0.20	0.17	no
CHSTUDUR1	0.62	0.54	no
CHSTUDUR2	0.28	0.29	no
U_CATHDUR	3.37	5.27	yes

Table 3.16 : Testing the mean for numerical variables grouped by DEATH (POSTOP group).

Variable Name	Mean value	Mean value	Sign. Diff.
	(DEATH = 0)	(DEATH = 1)	(alpha = 0.05)
EPIDU_DUR	1.32	1.08	no
C_LINE_NUM	0.51	0.86	yes
AGE	62.38	70.95	yes
APACHE2	12.47	17.97	yes
DAYS_ICU	1.23	4.37	yes
DAYS_CNC	1.37	1.67	yes
DAYS_FLR	1.04	0.68	yes
DIS_DELAY	0.15	0.10	no
ARTLIN_NUM	0.76	0.92	yes
ARTLINDUR1	2.27	2.85	yes
ARTLINDUR2	0.21	0.91	yes
ARTLINDUR3	0.03	0.16	yes
PACATH_NUM	0.27	0.35	yes
PACATHDUR1	0.72	0.96	yes
PACATHDUR2	0.07	0.23	yes
PACATHDUR3	0.03	0.20	yes
C_LINEDUR1	1.75	2.67	yes
C_LINEDUR2	0.13	1.00	yes
C_LINEDUR3	0.06	0.50	yes
VENT_NUM	0.39	0.66	yes
VENTHRS	23.14	79.30	yes
CHSTUBNUM	0.30	0.36	no
CHSTUDUR1	0.85	1.03	no
CHSTUDUR2	0.40	0.62	no
U_CATHDUR	3.37	5.95	yes

Table 3.17 : Testing the mean for numerical variables grouped by DEATH (NON-POSTOP group).

Variable Name	Mean value	Mean value	Sign. Diff.
	(DEATH = 0)	(DEATH = 1)	(alpha = 0.05)
EPIDU_DUR	0.14	0.05	no
C_LINE_NUM	0.31	0.63	yes
AGE	53.22	67.31	yes
APACHE2	12.36	21.57	yes
DAYS_ICU	1.77	3.77	yes
DAYS_CNC	1.86	1.02	yes
DAYS_FLR	0.97	0.49	yes
DIS_DELAY	0.22	0.10	yes
ARTLIN_NUM	0.42	0.67	yes
ARTLINDUR1	1.44	2.20	yes
ARTLINDUR2	0.33	0.55	no
ARTLINDUR3	0.08	0.16	no
PACATH_NUM	0.08	0.21	yes
PACATHDUR1	0.26	0.61	yes

PACATHDUR2	0.12	0.13	no
PACATHDUR3	0.03	0.02	no
C_LINEDUR1	1.07	2.05	yes
C_LINEDUR2	0.24	0.70	yes
C_LINEDUR3	0.21	0.59	no
VENT_NUM	0.29	0.74	yes
VENTHRS	43.55	71.26	yes
CHSTUBNUM	0.07	0.06	no
CHSTUDUR1	0.33	0.25	no
CHSTUDUR2	0.13	0.09	no
U_CATHDUR	3.36	4.87	yes

It has been found that there is a certain (less than moderate) negative relationship (the calculated Phi Coefficient is -0.141) between POSTOP and DEATH. In other words, the mortality rate of a patient after surgery is a little lower than a patient without surgery in the ICU. Figure 3.10 is part of the output result from running the SAS program class.sas using the FREQ procedure with options CHISQ and MEASURES.

```
TABLE OF POSTOP BY DEATH

POSTOP     DEATH
Frequency Percent
Row Pct
Col Pct    0       1      Total
------------------------------------
   0      1210    299    1509
          37.67   9.31   46.98
          80.19   19.81
          44.08   64.03
------------------------------------
   1      1535    168    1703
          47.79   5.23   53.02
          90.14   9.86
          55.92   35.97
------------------------------------
Total     2745    467    3212
          85.46   14.54  100.00            ( Frequency Missing = 18)

       STATISTICS FOR TABLE OF POSTOP BY DEATH

Statistic                  DF       Value      Prob
------------------------------------------------------
Chi-Square                 1        63.742     0.001
Phi Coefficient                     -0.141
```

Figure 3.10: General association between POSTOP and DEATH.

The chi-square test was also used to test the relationship of other nominal variables in the DECH ICU data set. Basically, for nominal variable 1 and nominal variable 2, the null hypothesis is that "these two variables are not associated", and the alternative hypothesis is that "these two variables are associated". After testing, we found that if we set the type I error rate alpha to be 0.05, then (i) there is significant association between POSTOP and SEX, and the calculated statistics indicate that in the POSTOP group, the proportion of male patients over total patients is a little larger than that in the NON-POSTOP group, which means that we accept the alternative hypothesis; (ii) there is no significant association between POSTOP and ARTLIN_COM; (iii) there is no significant association between SEX and ARTLIN_COM; (iv) there is no significant general association between DEATH and ARTLIN_COM; and (v) there is no significant association between DEATH and SEX. (ii) - (v) means that in those cases, we cannot refuse the null hypothesis. Table 3.18 illustrates more details of this testing information. In Table 3.18, DF represents the degrees of freedom of a Chi-Square distribution, and the negative Phi coefficient indicates the relation of two variables is somewhat negative. For example, the proportion of patient death over total patients who had EPIDUR is a little smaller than the proportion of death patients over total patients who did not have EPIDUR. The positive Phi coefficient indicates that the relation between two variables is somewhat positive.

Table 3.18 : General association between two nominal variables in DECH ICU data set.

Variable 1	Variable 2	DF	Chi-square Value	Prob.	Significant Association (alpha = 0.05)	Phi-Coefficient
POSTOP	ARTLIN_COM	1	2.826	0.093	No	-0.035
POSTOP	NG_TUB	1	28.505	0.001	Yes	0.095
POSTOP	EPIDUR	1	576.033	0.001	Yes	0.436
POSTOP	SEX	1	14.061	0.001	Yes	0.066
SEX	ARTLIN_COM	1	0.193	0.661	No	0.009
SEX	NG_TUB	1	1.333	0.025	Yes	0.021
SEX	EPIDUR	1	21.476	0.001	Yes	0.085
SEX	DEATH	1	0.187	0.666	No	0.008
ARTLIN_COM	DEATH	1	2.111	0.146	No	0.030
NG_TUB	DEATH	1	33.653	0.001	Yes	0.103
EPIDUR	DEATH	1	27.834	0.001	Yes	-0.096

It has also been found that for most numerical clinical parameters, for both the POSTOP group and the NON-POSTOP group, the mean value of the observations for arterial line catheter complication patient group (ARTLIN_COM = 1) is significantly different from that in the other group (ARTLIN_COM = 0). The test was done using the SNK method which is commonly used to compare pairs of means.

It has also been found that for most numerical clinical parameters, for both the POSTOP group and the NON-POSTOP group, the mean value of the observations in the male group (SEX = M) is not significantly different from that in the female group (SEX = F). The test was done using the SNK method.

Chapter 4

MODELING THE ICU PATIENT

4.1 Introduction

Patients with a critical illness go through a number of stages between being admitted to being discharged. Data describing their conditions comes from a variety of sources such as admitting diagnosis, on-line heart rate monitors and APACHE II scores. These data and the traditional medical model (history or physical examination and evaluation of the usual laboratory testing) are used by physicians to make clinical decisions. It has been found [Civetta, 1993] that the same parameters can play different roles at different times during a patient's stay in the ICU. For example, it is not unusual that, for an ICU patient admitted from an emergency room, parameters related to cardiorespiratory integrity are much more weighty than others since death is probably the most important factor to be considered at that time. Physicians are especially attentive to cardiorespiratory parameters during this period. As time goes on, for a long-term ICU patient, other parameters become dominant.

If a decision needs to be made or a framework for plans for the future needs to be formed, then some old information along with new information must be used. In other words, clinical decision making must be a process of many steps based on elements gathered over time. To increase the reliability of a case-based reasoner in the different ICU phases, different parameter weights are considered for use for the ICU patient in different phases of the ICU. Consequently, modeling the ICU patient becomes a necessity. At this point, Civetta's model [Civetta, 1993] can be used to guide the modeling effort.

4.2 Clinical Decision Making - Influence of Time

Civetta [1993] proposes a model of **Clinical Decision Making - Influence of Time** in his book ***Critical Care Medicine***. In this book, Civetta proposes that the basis of fundamental clinical decisions are contained in the elements of the clinical care process and components of the predictive indices such as the APACHE score, heart rate and trauma score. This model suggests that the clinical care process can be viewed from six separate phases or vantage points during the passage of time to assess the objectives of therapy and the results attained. These six phases are source of admission, on ICU admission, short-term ICU outcome, continued ICU care, long-term ICU patients and discharge from intensive care. At each temporal vantage point, examining the predictive indices or scoring systems to extract the important elements is very helpful to increase the correctness of clinical decision making. Combined with the elements of the clinical care process, we can focus our attention on the relevant physiologic processes that have the greatest effect on survival at that vantage point. In this way, clinical decision-making is concerned with appropriateness of therapy, given the existing circumstances for this patient at this point in the illness. Figure 4.1 shows the diagram of Civetta's model.

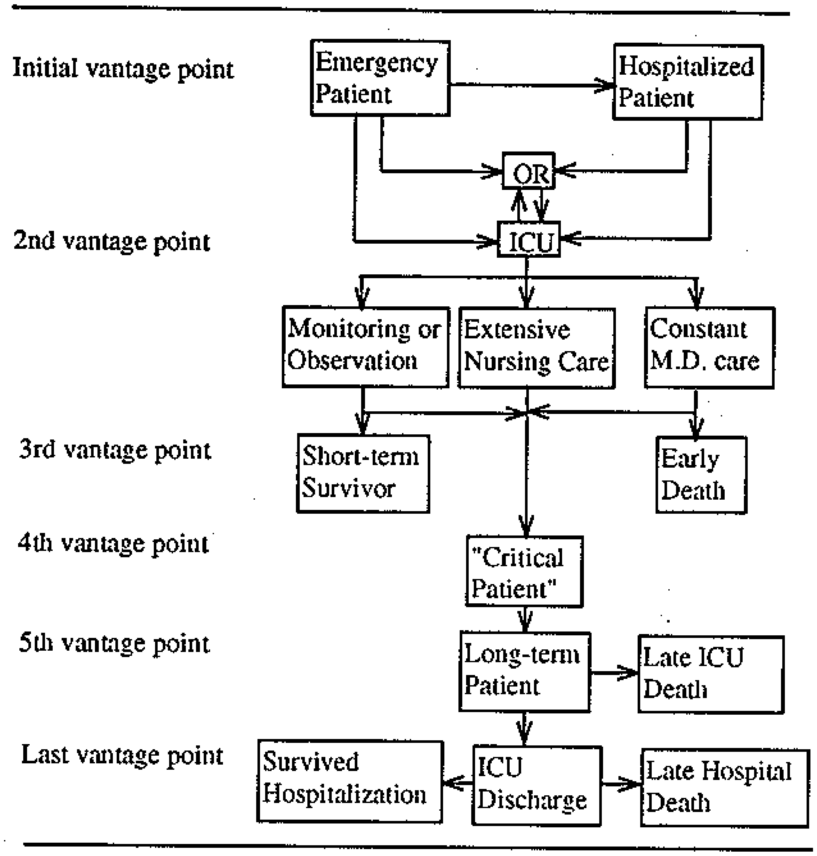


Figure 4.1: Clinical Decision-Making - Influence of Time, from Civetta [1993].

4.3 Modeling the ICU Patient of DECH

As each phase in Civetta's model has its own special characteristics, different approaches to clinical decision making should be made in each phase. This section presents discussion of each phase of Civetta's model taking into account the special circumstances of the DECH. The different information about the importance of parameters of the ICU patient database of the DECH at each vantage point is obtained. Some statistical results described in Chapter 3 were used to assist in the analysis.

The first phase is concerned with source of admission. In Civetta's model, this phase describes two main sources: emergency and elective. This is consistent with the ICU at the DECH. According to Civetta's model, for emergency patients the crucial parameters of clinical decision making in this phase are related to cardiorespiratory integrity, and for elective patients (usually surgical postoperative), the capacity to withstand future physiologic stress is the most important factor. According to Civetta [1993] and Vij et al [1981], in the evaluation of the cardiovascular system of a critically ill patient, the central venous pressure (CVP), pulmonary capillary wedge pressure (PCWP), pulmonary and systemic vascular resistance, heart rate, height, and weight are important. Unfortunately, these parameters are not shown in the ICU database of the DECH. The important parameters, or leading parameters, in the DECH ICU database in this stage may be considered as AD_DX1 (admitting diagnosis no. 1), along with the patient AGE.

The second phase is on ICU admission. Since a presumed increased risk of death is a common factor of the patient who is admitted to the ICU, examining the factors correlated with mortality will be a necessity at this vantage point. Some measurement common to critically ill patients can be used. For patients admitted from surgery, again, cardiorespiratory function and oxygen transport are the most important components, and for other patients without acute physiologic mental confusion, parameters reflecting the state of the immune system have the best predictive value. According to Vij et al [1981], some parameters such as the mixed venous oxygen tension (Pvo_2), serum lactate level, and colloid oncotic pressure (COP) can be good prognostic indicators in the evaluation of a critically ill patient on aspects of oxygen uptake, transport, and delivery. The most important parameter in the ICU database of DECH in this phase is AD_DX1 (admitting diagnosis no. 1). Depending on its value (POSTOP or not), two different patient groups will be classified. From Chapter 3, we know that approximately two-third of patients having an admitting diagnosis of POSTOP in the ICU of the DECH.

The third phase (0-24 hours approximately) is about short-term ICU outcome. When a patient is admitted to the ICU, we must distinguish which group this patient belongs to. In Civetta's model three groups are identified. These three groups are short-term survivor, early death and "critical patient". The first and second groups are short-term ICU admission groups and the last group will have a long-term stay in ICU. The relationship among mortality rate, duration of ICU stay and severity of illness, is, in fact, quite complex. It is usual that the mortality rate is proportional to the duration of the ICU stay. However, long ICU stay is not necessarily correlated with a higher severity of illness even though the term "critical patient" is used to represent the long-term admission ICU patient in Civetta's model. Civetta suggested that in this phase, the most important factor is to distinguish a short-term survivor from early death. APACHE, TISS, CRV and MLR are very useful systems to assist in making this determination. In the ICU of the DECH, the APACHE II system is used and it is compressed to 12 routine physiological measurements plus age and previous health status. APACHE2 is a parameter (a field) in the ICU database of the DECH. The APACHE II score employs physiological assessments that are based on the worst values during the first 24 hours in the ICU. The DECH ICU uses the values at the time of admission which is different from the literature standard.

The fourth phase (24 hours - two weeks approximately) is on continued ICU care. In general, clinical decision making is very difficult at this phase. Nobody knows exactly what will happen. The mortality rate of this phase is quite high (approximately 50%), but only time can tell who is going to die and who is going to survive. Intervention seems always necessary. At this phase, clinical decision making is not easy. Collecting all information is valuable. Based on this, we may conclude that parameters such as ARTLINDUR1 (arterial line catheter duration no.1), VENTHRS (number of hours of ventilated) and DAYS_ICU (days in ICU) in the ICU database of DECH can be considered to be very important.

The fifth phase (more than two weeks, approximately) is for long term ICU patients. In this phase, the most important thing is to try to discern eventual survivors from occurrence of late death. Methods of measuring protein metabolism and immune function can assist in decision making in this phase. Some parameter values can show significant differences between surviving and dying patients. For example, Civetta shows that Urea and Lactate show differences between surviving and dying septic patients. None of those parameters are included in the ICU database of the DECH. It is worth mentioning that most of the parameters that have been found to be discriminatory in phase five patients are not related to acute cardiorespiratory or to the types of bedside assessment which is very valuable for early ICU clinical decision making. VENTHRS (number of hours of ventilated) and DAYS_ICU (days in ICU) can be considered important in this phase.

The sixth or last phase concerns the discharge from the ICU. At this stage, a decision has to be made carefully. Since many things could happen after the patient is discharged (for example, late hospital death or a permanent stay in a nursing home), social parameters must be considered in clinical decision making at this phase. It is very hard to say but it may be true that preserving the quality of life may be the most important thing. Considering the ICU patient database of the DECH, AGE and CHR_DX* (chronic diagnoses) can be considered as leading parameters for our case-based reasoner.

The above six phase classification is a very detailed patient classification in the ICU. The basic idea of this classification was incorporated into IDEAS for ICUs Version 3.0. A slight modification is that we decided to consider three phases instead of six when we designed our case-based reasoner considering our specific situation. These three phases are based on time. The first phase is during the first 24 hours (short-term ICU outcome); the second phase is during the first two weeks (continued ICU care); and the third phase is at any time longer than two weeks (long-term ICU patients). This is reasonable due to the facts that (i) we don't have as many parameters as those discussed in Civetta's model, and (ii) two databases ICU93_P.dbf and ICU93_O.dbf (see Chapter 5) which are subsets of ICU93.dbf were used to build the case bases.

Chapter 5

ADAPTING AN ICU MEDICAL DATABASE TO A CASE-BASED SYSTEM

5.1 Constructing Case Bases from a Medical Database

A case base in a Visual Basic application (using a modified VBXpert custom control) can be represented by an external database file with some fields indexed using some special functions of The Easy Reasoner DLL CBS2WS.DLL. The number of cases in the case base is exactly the number of records in the database file. A weight (if it is not the default 1.0) must be set for a field before building an index associated with this field. Different weights for the same field can be set only if the previously built index associated with this field has been destroyed.

The time to build a case base index is typically proportional to the size of a database file. At the present time, the use of computers in health care has become increasingly common. If a CDSS such as a case-based reasoner wants to be a leading product, one of the most important things is that it must be fast. In other words, it must not only be effective but also efficient enough to meet the user's expectation to a large extent. The DECH/ICU database file ICU93.dbf has more than three thousand records and the size will probably increase dramatically as time goes on. A very natural idea is to break ICU93.dbf into several database files. For each smaller database file, depending on the requirement, one or more case base indices can be built. A key problem is how to decide the database and case base architecture and the rationale.

5.1.1 Rationale

Based on the statistical analysis in Chapter 3 and the DECH/ICU patient model in Chapter 4, the weights of a clinical parameter (a field in the ICU database) are different in each phase of the ICU stay. In addition, we know that the patients in the POSTOP group have many characteristics that are different from the NON-POSTOP group, including different mortality rates and a different importance of the clinical parameters in each phase of the ICU. In the ICU, clinical decision making is quite different for these two patient groups. The weights of clinical parameters need to be considered separately for these groups. The relationship within each group is much closer than that in the mixture for some parameters. Due to the above facts, the ICU93.dbf was split into two database files. For each database file, based on the weights of each clinical parameter in different phases of the ICU, three case base indices were built. It is anticipated that this classification will increase the interdependence between a new case and a matched case.

5.1.2 Database and Case Base Architecture

Two database files were formed based on the value of AD_DX1 which is a field of the DECH/ICU database ICU93.dbf. One is for patients after surgery and the other one is for other patients. If the value of AD_DX1 for a patient record happens to be "POSTOP", then it is classified as database file I. If the value for the patient record of AD_DX1 happens to be other, then this record is classified as database file II. We named the database file I as ICU93_P.dbf and the database file II as ICU93_O.dbf. There are some relationships between the database and the case base index. According to the DECH/ICU patient model in Chapter 4, three case bases (case base indices) were constructed for each of these two database files. Time is the main factor considered here. Case base I focuses on the first 24 hours of patients in the ICU. Case base II focuses on the first two weeks of patients in the ICU. Case base III focuses on patients staying longer than two weeks in the ICU. Figure 5.1 shows the database and case base architecture.

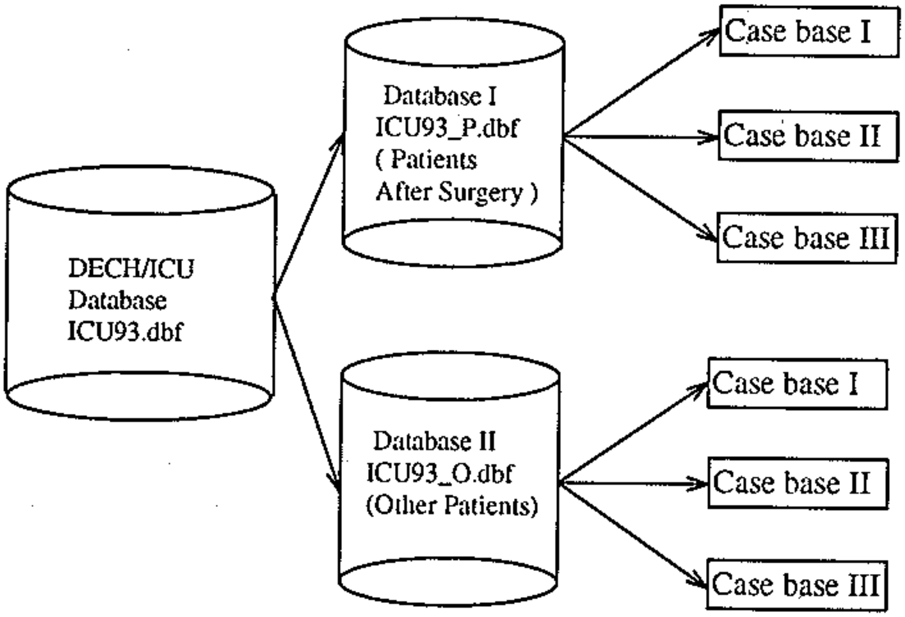


Figure 5.1: Database and case base architecture.

5.1.3 Updating a Case Base

As we discussed above, a case-based reasoner such as our application IDEAS for ICUs Version 3.0 is supposed to be used to assist physicians in making clinical diagnoses and making decisions on patient management. The way our case-based reasoner works is "given a new case, find the top ten closest matching cases in the case base." As a result, for our application, the top ten closest patient records (the most similar ones to the new patient) from the ICU database are displayed on the screen. Once again, the case base index for the ICU database plays a large role in the matching process. The case base index can be either the one previously built or a new one (updated one) built at run time. It is expected that most users will just load the previously built index and perform queries they have in hand without bothering to refresh the case base index.

At some specific times, building a new index may become really necessary. The main reasons for rebuilding a case base index are because databases are updated, or field weights are tuned. Even though the number of cases in the case base is fully dependent on the number of records in the database, the matching result for a query depends on other things as well. The top ten closest matched cases not only depend on the records in the database, but also depend on other factors. These factors include the index field type, index field weight and other field features with some specific options that must be set before building a case base index (further details are in Chapter 2). In addition to the record number and the content of each matched case, the distance between the new case and each matched case needs to be retrieved based on the following rule: "the smaller the distance, the closer the match between the old case and the new one." There is no doubt that the previous similar cases can be good references for the physician to make correct clinical decisions. It also seems likely that no matter how much information the ICU database contains and how many patient records there are, it certainly cannot contain all the possible types of patient cases. There are more than three thousand patient records in the ICU database. It is still possible to perform a query to search for the top ten matching cases in a case base, only to find even the smallest distance d between the presented case and the matched cases being larger than 0.9 (or other large value α). What does this mean? Except for the misuse of the software, it may simply indicate that the new patient is a very different case. In some sense, finding a case with significantly different characteristics may be very important for clinical research. It may lead to finding a new disease or other significant events. This can be a good by-product of our case-based reasoner. At the time this "new case" is found, we need to add its record into the database and rebuild the case base index. It can be very valuable for future decision making. At least, there is an "exemplar" now. When a similar case is encountered in the future, the previous similar instance of it can be considered.

A new case will be added into ICU93.dbf. Depending on the situation, this new case may also be added into one of ICU93_P.dbf and ICU93_O.dbf. The principle used here is that if d - the minimum value of the distances associated with the top ten matching cases - is smaller than α, then the new case will be added into ICU93.dbf only. If d is larger than or equal to α, then the new case will be added into both ICU93.dbf and ICU93_P.dbf if the patient is admitted following surgery, or add the new case to both ICU93.dbf and ICU93_O.dbf otherwise. The determination of α should be based on the opinion of physicians. The main idea is illustrated in Figure 5.2.

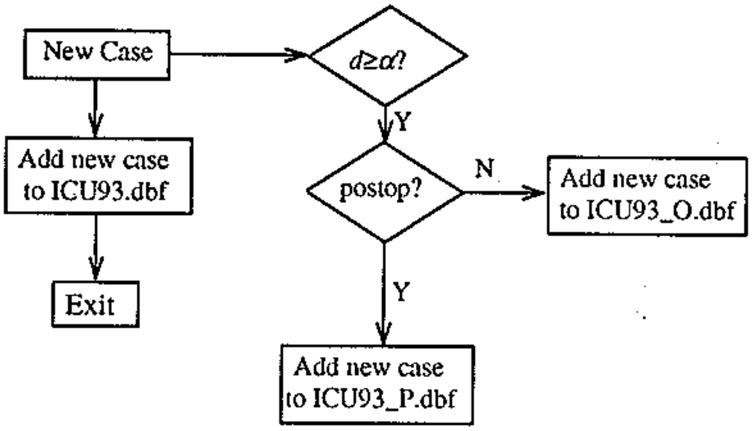

New Case
d≥α?
Y
Add new case to ICU93.dbf
postop?
N
Add new case to ICU93_O.dbf
Exit
Y
Add new case to ICU93_P.dbf

Figure 5.2: Updating databases with a new case.

5.2 Weight Determination

An ICU database usually contains many fields (or clinical parameters). As discussed above, depending on the situation (patient after surgery or not) and the time (different phase in the ICU), some fields may be considered more important than others. The more important the field, the higher the weight of that field. Before we build a case base index using some fields of the database, the information of the field weights must be translated into the type of weight The Easy Reasoner can understand. It is known that the weight of each field will heavily affect the matching result for a query. In other words, except for the presented case itself, the weights of each field are very important for deciding what the top ten match cases are for a presented case in the case base.

5.2.1 Primary Knowledge

In Chapter 3, some relationships were found among the clinical parameters (fields) in DECH/ICU database ICU93.dbf. These relationships were determined from Spearman correlation analysis, chi-square analysis and ANOVA (analysis of variance). In our application, the Spearman correlation coefficients and other statistics are considered to be very important factors to decide the weights of each field taking into account the specific characteristics of the three different phases of the modified ICU patient model.

5.2.2 Match Weights and Their Determination

For ease of use, a 0-100 scale was adopted for the weights. The 0-100 scale is used in the discussion in the following paragraphs and is also used in displaying the weights on the screen when running an application. The reason for doing this is that it seems that human beings always feel more comfortable dealing with the larger numbers such as 20 and 30 than dealing with smaller numbers such as 0.2 and 0.3.

To determine the weights for each clinical parameter (field) at different phases (see Chapter 4), we need to use the Spearman correlation coefficients and other statistics we obtained through the previous data analysis. The basic idea of weight determination is that, in each phase, the weights of the leading parameters are set to be 100. For other parameters used in the case base index, depending on the Spearman correlation coefficients and other statistics, their weight is set to be lower than the leading parameters. For example, if the Spearman correlation coefficient between a parameter and the leading parameter in that phase is 0.73, then the field weight of this parameter is set to be 73. If more than one leading parameters exists, then the maximum value of correlation coefficients between that parameter and the leading parameters is selected. Chapter 6 shows the detailed field weights in different case base indices.

Chapter 6

IMPLEMENTATION AND EVALUATION

6.1 The Graphical User Interface

Microsoft Visual Basic 4.0 is used as the graphical development environment for building IDEAS for ICUs version 3.0 mainly for three reasons. The first reason is simply because the previous version of IDEAS for ICUs used this environment. The second reason is that Visual Basic can be connected to The Easy Reasoner which we thought to be a better case-based reasoning tool than ART-IM and others. The third reason is about Visual Basic itself. Visual Basic is indeed a very convenient and productive GUI programming environment.

Visual Basic provides an intuitive screen painter used to paint windows with a variety of controls (check boxes, list boxes, push buttons, and so on). Each control has properties such as background color and starting text that can lay out the foundation for a single- or multiple-windowed Graphical User Interface (GUI). In addition to the programming features described above, Visual Basic has many other capabilities. The term "Basic" does not imply that its power is somehow limited. In fact, its power is much more than adequate to control the flow of an application. One of the capabilities of Visual Basic is that it provides access to dynamic link library (DLL) functions written in such languages as C and C++ which allow more complicated tasks to be performed. This capability is required for our application.

6.2 Computer Implementation of Distance Calculation

Chapter 2 illustrates how to calculate the distance between a query and a record. Based on the distances we calculated for each record, we can retrieve closest matching cases for the new case. This is the basic idea of how the case-based reasoner performs matching.

For the real computer implementation of distance calculation, working experience shows that two factors are important. The first factor is that if the pre-built indices are used, then the database files ICU93_P.dbf and ICU93_O.dbf must be kept as they are. They cannot be modified; otherwise, the pre-built indices will not be valid anymore. If the database files are modified, then some unpredictable results appear. If the two databases really need to be modified, then the case base indices must be rebuilt at each phase (see Chapter 5). The second thing is that given a *case* and a *distance*, the DLL function cbr_build_query (CBR_CASEP *case*, DISTANCE *distance*, NUMBER *atmost*, CBR_QUERYP **queryp*) (in CBR2WS.DLL) builds a query and places *atmost* cases in the handle returned in the reference argument *queryp*, where the *distance* should be set at a value just above the value we prefer, if we are guaranteed to retrieve *atmost* number of distinct cases. For example, if we think all records are acceptable as long as these records are the top ten matches, and the distances between the query and each retrieved record are less than or equal to one, then we should set the *distance* value to be a little more than one (e.g. 1.01). Our experiments have shown that if we use 1 instead of 1.01, then the top ten records are record 1 (the highest matching case) and record 2 (the second highest matching case) repeated nine times.

6.3 Integration with IDEAS for ICUs Version 2.3

In the existing user interface IDEAS for ICUs Version 2.3, the patient records are supposed to be stored in PATIENT.dbf, which was an empty database file before The Easy Reasoner, a case-based reasoning shell, was integrated. PATIENT.dbf has a slightly different table structure from ICU93.dbf. There are 130 fields in PATIENT.dbf (116 fields in ICU93.dbf) and most of them are either character fields or numeric fields. The contents of most fields in PATIENT.dbf are consistent with those of the fields in ICU93.dbf, even though the field names are defined in a slightly different way in the two database files. For example, admitting diagnosis no.1 is defined as AD_DX1 in ICU93.dbf but defined as ADDX1 in PATIENT.dbf. What we are most concerned about are the non-overlapping fields. In other words, some fields are defined in PATIENT.dbf but not in ICU93.dbf and vice versa. Figure 6.1 shows the details of these differences.

Fields in PATIENT.dbf but not in ICU93.dbf	**Fields in ICU93.dbf but not in PATIENT.dbf**
KEYVALUE	CCU
CCUOFLOW	EM-SURG
DAYSTOT	RF-MD
PERIPHIV	OTDEATHTIM
PERIPHDUR	O-PROCEDUR
PERIPHCOM	
AUTOSYRE	
HGT-CM	
WGT-KG	
BSA	
MED-1 .. 8	
STUDYREP	

Figure 6.1: Non-overlapping fields of PATIENT.dbf and ICU93.dbf.

The above discussion illustrates the need to build a new database file that contains all information from both database files. Since the table structure of PATIENT.dbf was defined later than that of ICU93.dbf, the former is used as the basis for the new file. The fields CCU, EM-SURG, RF-MD, O-PROCEDUR are included in the PATIENT.dbf table structure since they contain valuable information. For ease of implementing, PATIENT.dbf is used as the name of our new database file. All ICU93.dbf data is included in this newly defined database. Subsequent references to PATIENT.dbf refer to the new file unless specified otherwise. The new PATIENT.dbf file has 134 fields and 3300 patient records.

To achieve a better case matching result and also to make a contribution to any future research that may use PATIENT.dbf, some data cleaning was done for PATIENT.dbf. Some mistakes were corrected. A total of 169 values in PATIENT.dbf have been modified. Some examples of modifications are that in record 2502, at field ADDX1, TAUMA was changed to TRAUMA, and in record 3184, at field SEX, N was changed to M. The SAS program class.sas (see Appendix III) found the mistakes. For the details of data cleaning, one can refer to Chapter 3. ICU93_P.dbf and ICU93_O.dbf still retain the same name when the case-based reasoner was being developed, but with the table structures modified. The table structures of the ICU93_P.dbf and the ICU93_O.dbf are the same as those for PATIENT.dbf. ICU93_P.dbf contains 1712 patient records and ICU93_O.dbf contains 1518 patient records. ICU93_P.dbf and ICU93_O.dbf are obtained by running a simple Visual Basic program called divide.vbp on PATIENT.dbf.

6.4 Determining the Case Base Indexing Fields

Now, three database files PATIENT.dbf, ICU93_P.dbf and ICU93_O.dbf are obtained. They are the basis on which to build case base indices (see Chapter 5). In these three databases, 134 fields exist for each record. However, due to the facts that (i) some fields are either empty or only have a few observations, and (ii) some fields have little relationship with the leading parameters (see Chapter 4), it is not necessary to use all of them when building case base indices. To improve the efficiency of the case base reasoner, selecting some significant fields to build case base indices is a "must".

How are the indexing fields for our case bases chosen? Primarily, the output from SAS program class.sas was relied on. By running class.sas, a lot of first hand information such as frequency counts for all fields (except for patient names), descriptive statistics for all fields with numeric data type, Pearson correlation coefficients and chi-square analysis results (whenever they are applicable), was obtained. The basic rules to decide case base indexing fields can be described as follows: (i) if a field (excluding the leading parameters from Civetta's model, see Chapter 4) has a number of observations less than half of the entire records in a database file, then this field will not be used to build a case base index; (ii) if a field (excluding the leading parameters from Civetta's model, see Chapter 4) with 80 percent (for complication fields, this value is raised up to 85 percent) of observations being of the same value, then this field will not be used to build a case base index; (iii) if the Pearson correlation coefficients (if it can be calculated) of a field with all leading parameters (if applicable) is less than 0.05, then this field will not be used to build a case base index. Using these criteria, we have established that the fields shown in Table 6.1 are used to build case base indexes. In Table 6.1, the first column represents the order of indexing fields of case bases, the second column represents the indexing field name, the third column represents the indexing field type, the fourth column represents the number of observations found in that indexing field, the fifth column represents the percent of observations in that field being of the same value (must be the value repeated most frequently), and the sixth column adds comments.

Table 6.1 : List and information of case base indexing fields.

No.	Field Name	Field Type	No. of Obs.	% of Obs. are Same	Notes
1	SEX	C	3187	60.53	
2	EPIDUR	N	2769	74.7	
3	CENTNUM	N	2787	61.6	
4	AGE	N	3230	3.2	* in phase 1 (for POSTOP only), 3
5	APACHE	N	3224	48.9	* in phase 1
6	ADDX1	C	3228	53	* in phase 1 (for NON-POSTOP only)
7	ADDX2	C	3036	5.96	
8	ADDX3	C	2489	5.58	
9	CHRDX1	C	412	18.20	* in phase 3
10	DAYSICU	N	3167	58.3	* in phase 2, 3
11	DAYSCNC	N	3218	42.4	
12	DAYSFLR	N	3215	64.4	
13	ARTLINE	C	3214	53.45	
14	ARTNUM	N	2992	52.1	
15	ARTDUR1	N	2996	42.6	* in phase 2
16	ARTCOM	C	2281	84.52	
17	CENTLINE	C	3215	66.72	
18	CENTDUR1	N	2784	61.7	
19	VENT	C	3217	67.64	
20	VENTNUM	N	2787	62.8	
21	VENTDUR	N	2788	62.8	* in phase 2, 3
22	UCATH	C	3197	74.51	
23	UDUR	N	3099	24.4	
24	NGTUBE	C	3177	64.97	
25	EPIDURAL	C	3025	76.53	

Note: * represents leading parameter (see Chapter 4).

6.5 Evaluating the Justified Weights

The indexing field weights at each phase (See Chapter 4) are illustrated in Table 6.2. In Table 6.2, 'P' represents POSTOP and 'O' represents others, i.e., NON-POSTOP. The weights of SEX, NGTUBE and EPIDURAL at each phase were set to be 5 because (1) through a chi-square test, it was found that the general association between each of these fields and POSTOP is significant but less than moderate; (2) through the SNK test, it was found, only for a few fields, that the means of observation in one group are significantly different from the ones in the other group. The type I error rate Alpha was set to 0.05. So, 5 was believed to be a suitable value to describe the status of these fields in the case base index. The weights of some fields with binary values were decided based on the weights of some other associated fields. The fields associated with leading fields have higher weight that others. For example, the weight of VENT at each phase was decided based on the weight of VENTDUR. If the weight of VENTDUR at a phase happens to be the largest value among other phases, then we set the weight of VENT to be 75 for that phase, and 50 for other phases. We set the weight of any complication field such as ARTCOM to be 50. For the details of how to determine the weights of other fields, refer to Section 5.2.

Table 6.2 : Field weights at each phase for ICU93_P.dbf (P) and ICU93_O.dbf (O).

No.	Field Name	Phase 1 (P)	Phase 2 (P)	Phase 3 (P)	Phase 1 (O)	Phase 2 (O)	Phase 3 (O)	1994 weights
1	SEX	5	5	5	5	5	5	5
2	EPIDUR	12	37	16	10	18	9	3
3	CENTNUM	40	61	61	41	58	58	3
4	AGE	100	11	100	45	19	100	100
5	APACHE	100	57	57	100	52	52	80
6	ADDX1	N/A	N/A	N/A	100	80	60	100
7	ADDX2	80	60	40	80	60	40	100
8	ADDX3	80	60	40	80	60	40	100
9	CHRDX1	25	50	100	25	50	100	90
10	DAYSICU	53	100	100	52	100	100	85
11	DAYSCNC	9	32	32	22	33	33	N/A
12	DAYSFLR	8	14	14	16	15	15	N/A
13	ARTLINE	50	75	50	50	75	50	5
14	ARTNUM	27	78	45	37	95	68	8
15	ARTDUR1	37	100	56	34	100	70	3
16	ARTCOM	50	50	50	50	50	50	25
17	CENTLINE	25	50	50	25	50	25	25
18	CENTDUR1	43	66	66	41	60	59	3
19	VENT	50	75	75	50	75	75	60
20	VENTNUM	56	96	96	53	96	96	N/A
21	VENTDUR	57	100	100	48	100	100	80
22	UCATH	50	75	50	50	75	50	3
23	UDUR	38	59	55	33	66	65	3
24	NGTUBE	5	5	5	5	5	5	5
25	EPIDURAL	5	5	5	5	5	5	5

Table 6.2 also displays a part of the weights used in previous version of IDEAS for ICUs. The previous version of IDEAS used around 50 fields when building the case base. The new version of IDEAS only used 25 fields that have been studied and proved to be important. This is another significant difference between two versions of IDEAS.

6.6 IDEAS for ICUs Version 3.0

IDEAS for ICUs Version 3.0 was developed by integrating a new approach to case-based reasoning (using a case-based reasoning shell called The Easy Reasoner) using medical data, with an existing graphical user interface IDEAS for ICUs version 2.3. In general, comparing with IDEAS for ICUs Version 2.0 [Taylor, 1994], one of the characteristics of IDEAS for ICUs Version 3.0 is that there is a big improvement in the aspect of the running speed. Some test programs were written to obtain some detailed timing information. The machines we used to do the experiments is IBM PS/2 Model 90 XP 486 (33MHz, with 16 MB RAM running Windows 3.1). Given 2000 patient records (subset of DECH ICU data) in a database, we have measured the time that was spent on building a new case base index, loading an existing case base index, or finding the top ten matched cases in the case base for a new presented case, separately. The number of fields we used to build the case base indices are 50, 40, 30, 20 and 10, respectively. The field types of most fields in the case base indices are defined as numeric, or symbol and a few are defined as text. This is consistent with the case bases we used in the IDEAS for ICUs Version 3.0. Table 6.3 provides detailed information. Please note that the time shown in Table 6.3 does not include the time needed to show the summary output window. The time for both case matching and summary output window display is 57 seconds on average on the PS/2 Model 90. Using an IBM 350, Pentium 166MHz processor, 32 MB RAM, running Windows 95, the average time for both case matching and summary output window display is 8 seconds.

Table 6.3: Sample timing table associated with case base indices.

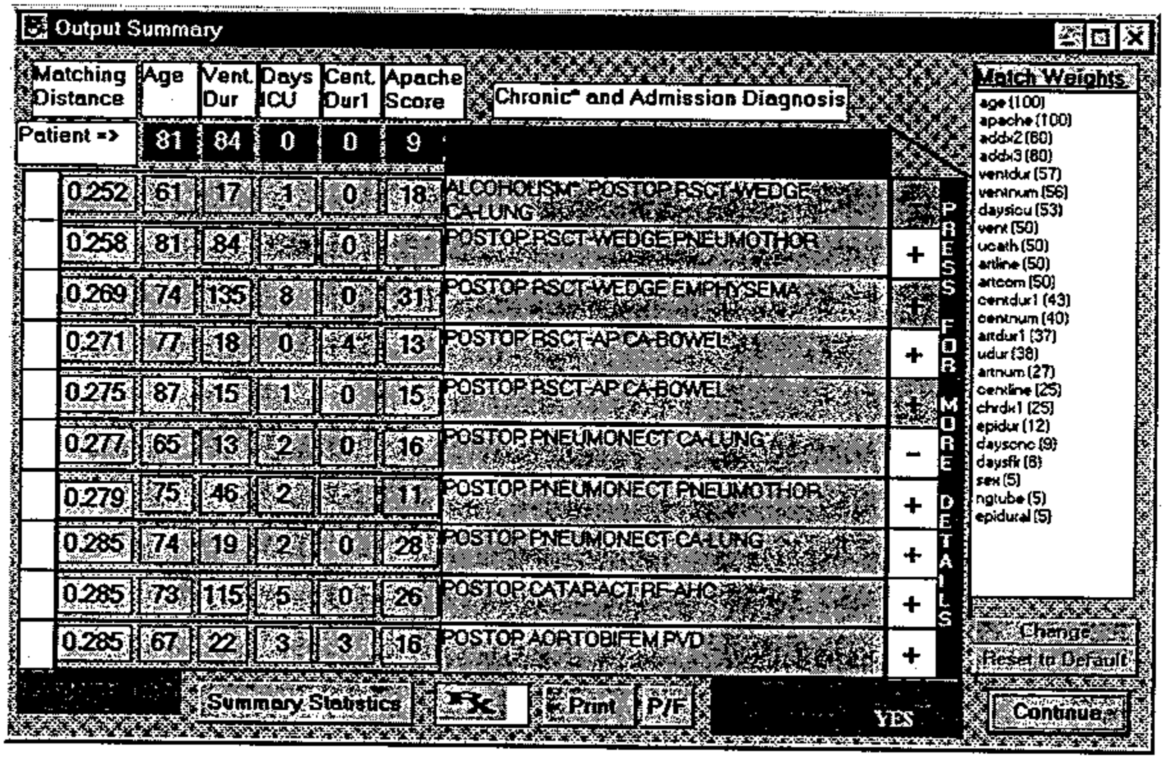
Number of Indexing Fields	**Building Index Time (sec.)**	**Loading Index Time (sec.)**	**Case Matching Time (sec.)**
50	35	3	6
40	34	2	5
30	34	2	4
20	29	2	4
10	22	1	1

During the experiments, we found that the time for building a case base index or finding the top ten matching cases in the case base for a new presented case with text fields is a bit longer than with numeric or symbol fields.

The second characteristic of IDEAS for ICUs Version 3.0 is that it is a very user friendly program. We know that, most of the time, the end user of IDEAS for ICUs Version 3.0 need not rebuild the case base indices. To find the top ten matches for a current patient in the ICU, the simplest way is to click the button "Display Matches" to bring all current patient names in the DECH ICU up in a list box, and then choose the patient to use as the presented case. Then, a summary output window is displayed. If the user wants to see more information on a particular matching patient, they click once on the rightmost '+' or '-' sign.

The third characteristic of IDEAS for ICUs Version 3.0 is that it shows the most important information in the summary output window. Since we have defined three different phases for the ICU and concluded that different leading parameters exist in different phases based on Civetta's model, the contents of the summary output window for each phase looks a little bit different. The idea is that we want to present the most important information to the physician as quickly as possible.

Figure 6.2 illustrates the top ten matching cases for a new presented case which belongs to the POSTOP group in the first phase (0-24 hours).

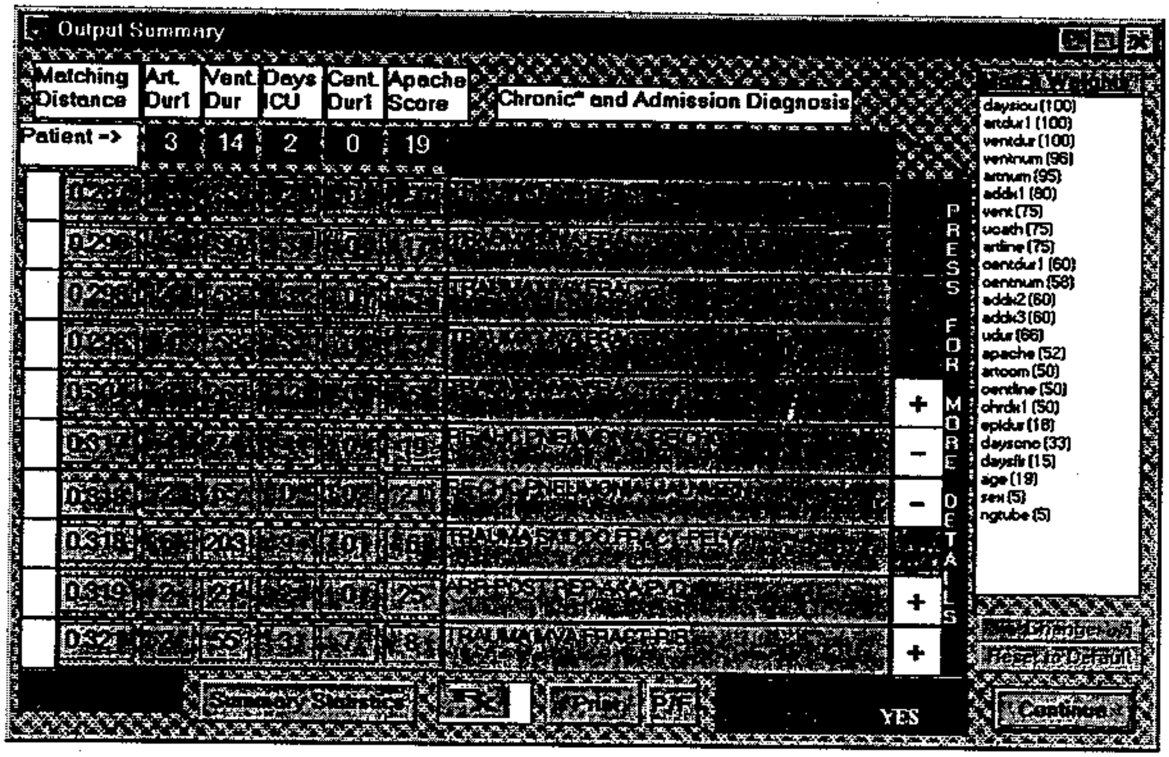


Matching Distance	Age	Vent. Dur	Days ICU	Cent. Dur1	Apache Score	Chronic* and Admission Diagnosis	
Patient =>	81	84	0	0	9		
0.252	61	17	1	0	18	ALCOHOLISM* POSTOP RSCT-WEDGE CA-LUNG	
0.258	81	84	[illegible]	0	[illegible]	POSTOP RSCT-WEDGE PNEUMOTHOR	+
0.269	74	135	8	0	31	POSTOP RSCT-WEDGE EMPHYSEMA	+
0.271	77	18	0	4	13	POSTOP RSCT-AP CA-BOWEL	+
0.275	87	15	1	0	15	POSTOP RSCT-AP CA-BOWEL	+
0.277	65	13	2	0	16	POSTOP PNEUMONECT CA-LUNG	−
0.279	75	46	2	[illegible]	11	POSTOP PNEUMONECT PNEUMOTHOR	+
0.285	74	19	2	0	28	POSTOP PNEUMONECT CA-LUNG	+
0.285	73	115	5	0	26	POSTOP CATARACT RF AHC	+
0.285	67	22	3	3	16	POSTOP AORTOBIFEM PVD	+

Match Weights
age (100)
apache (100)
addx2 (80)
addx3 (80)
ventdur (57)
ventnum (56)
daysicu (53)
vent (50)
ucath (50)
artline (50)
artcom (50)
centdur1 (43)
centnum (40)
artdur1 (37)
udur (38)
artnum (27)
cenline (25)
chrdx1 (25)
epidur (12)
daysonc (9)
daysfr (8)
sex (5)
ngtube (5)
epidural (5)

Figure 6.2: Top ten matching cases in the POSTOP group at the first phase.

Figure 6.3 illustrates the top ten matching cases for a new presented case which belongs to the NON-POSTOP group in the second phase (24 hours - two weeks).



Output Summary

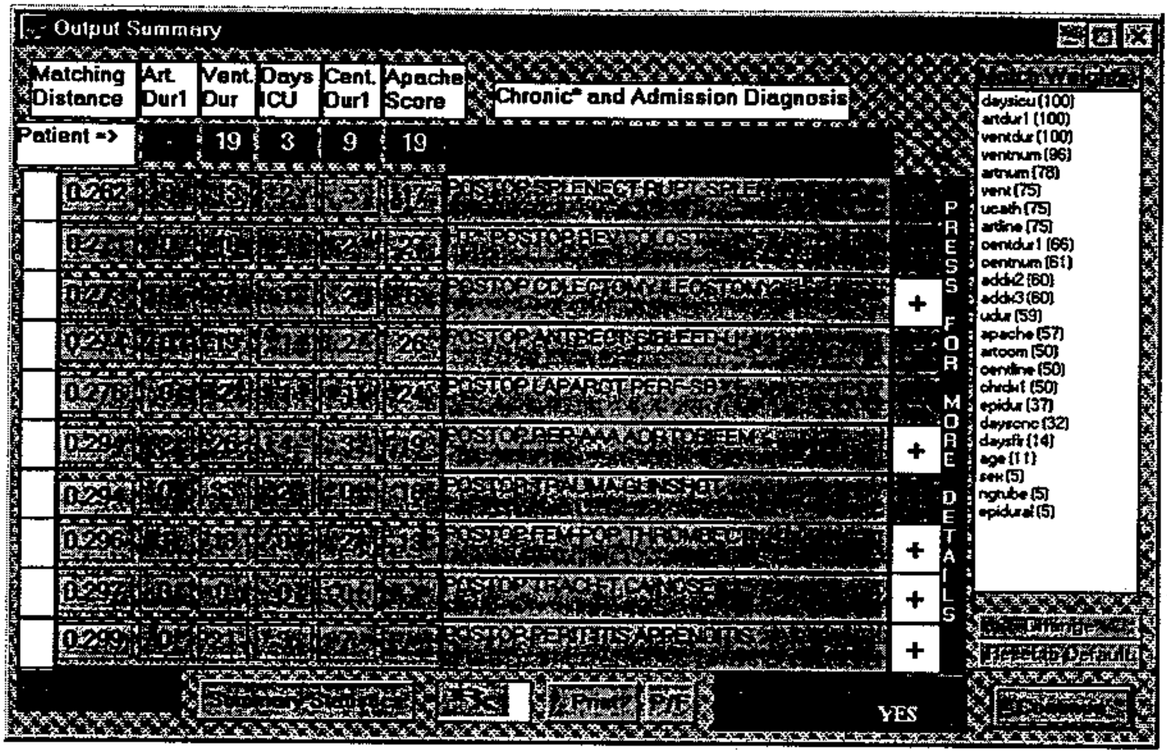
Matching Distance	Art. Dur1	Vent. Dur	Days ICU	Cent. Dur1	Apache Score	Chronic* and Admission Diagnosis
Patient ->	3	14	2	0	19	

daysicu (100)
artdur1 (100)
ventdur (100)
ventnum (96)
artnum (95)
addx1 (80)
vent (75)
ucath (75)
artline (75)
centdur1 (60)
centnum (58)
addx2 (60)
addx3 (60)
udur (66)
apache (52)
artoom (50)
centline (50)
chrdx1 (50)
epidur (18)
daysono (33)
daysfir (15)
age (19)
sex (5)
ngtube (5)

Figure 6.3: Top ten matching cases in the NON-POSTOP group at the second phase.

In Figures 6.2 and Figure 6.3, a "-" sign (which is close to "PRESS FOR MORE DETAILS" on the right) means that the matched patient died in the ICU and a "+" sign means that the matched patient was still alive at the time of discharge from the ICU. The shadow overlaid on the "+" or "-" sign indicates that the arterial line catheter complications field was not true for that patient. From the "Match Weights" column, we can find the relative weight of each field in the case base index. These weights were used to perform matching. The "Chronic* and Admission Diagnosis" column displays chronic admission diagnosis no.1 and admitting diagnosis no.1 to no.3. The first column close to the "Matching Distance" displays either "age" or "artdur1" depending on which one is more important at that specific phase.

Figures 6.4 and Figure 6.5 show the matching results under other situations. Figure 6.4 illustrates the top ten matching cases for a new presented case which belongs to the POSTOP group in the second phase (24 hours - two weeks).



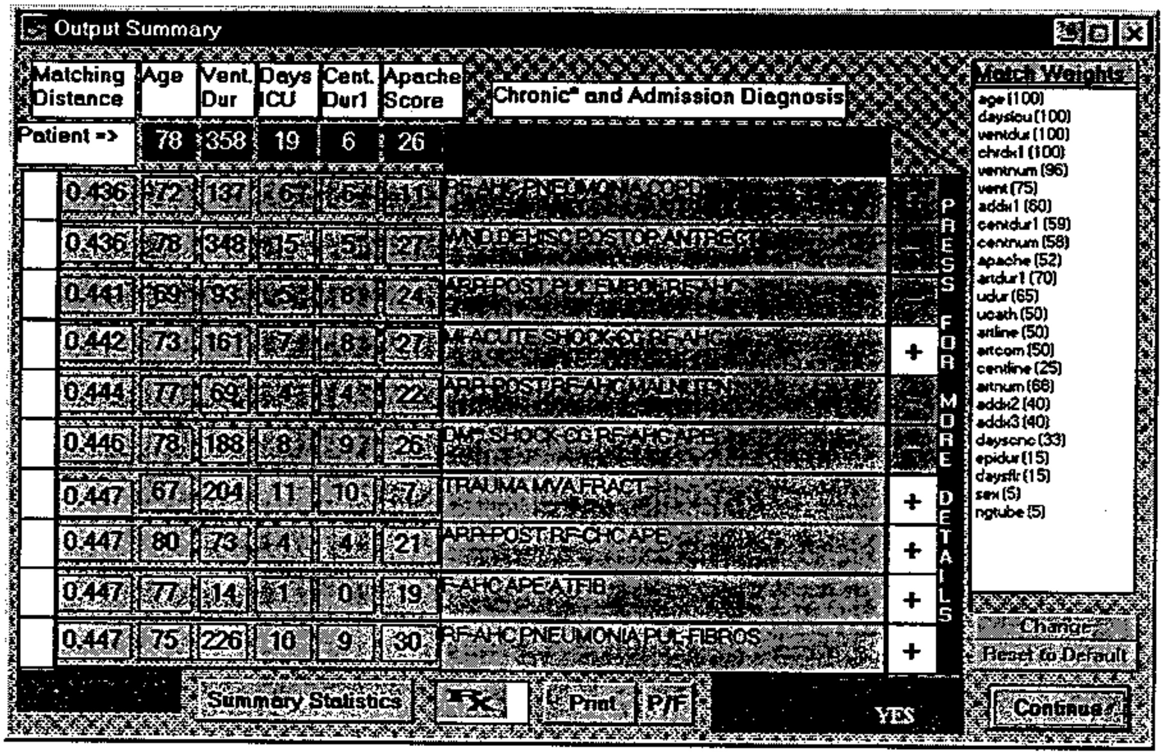
Output Summary

Matching Distance	Art. Dur1	Vent. Dur	Days ICU	Cent. Dur1	Apache Score	Chronic* and Admission Diagnosis
Patient =>	-	19	3	9	19	
0.262	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	POSTOP SPLENECT RUPT SPLE[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	POSTOP REV COLOST[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	POSTOP COLECTOMY ILEOSTOMY[illegible]
[illegible]	[illegible]	19	[illegible]	[illegible]	26	POSTOP ANTI BEST BLEED U[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	POSTOP LAPAROT PERF SB[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	POSTOP PER AAA AORTO FEM[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	POSTOP TRAUMA GUNSHOT[illegible]
0.296	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	POSTOP FEM POP THROMBECT[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	POSTOP TRACH T CA NOSE[illegible]
0.299	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	POSTOP PERITITIS APPENDITIS[illegible]

daysicu (100)
artdur1 (100)
ventdur (100)
ventnum (96)
artnum (78)
vent (75)
ucath (75)
artline (75)
centdur1 (66)
centnum (61)
addx2 (60)
addx3 (60)
udur (59)
apache (57)
artcom (50)
centline (50)
chrdx1 (50)
epidur (37)
dayscnc (32)
daysfr (14)
age (11)
sex (5)
ngtube (5)
epidural (5)

Figure 6.4: Top ten matching cases in the POSTOP group at the second phase.

Figure 6.5 illustrates the top ten matching cases for a new presented case which belongs to the NON-POSTOP group in the third phase (longer than two weeks).



Matching Distance	Age	Vent. Dur	Days ICU	Cent. Dur1	Apache Score	Chronic* and Admission Diagnosis	
Patient =>	78	358	19	6	26		
0.436	72	137	6	6	11	RF-AHC PNEUMONIA COPD	−
0.436	78	348	15	5	27	WND DEHISC POSTOP ANTREST	−
0.441	69	93	5	8	24	ARR-POST PUL EMBOL RF-AHC	−
0.442	73	161	7	8	27	MI ACUTE SHOCK-CG RF-AHC	+
0.444	77	69	4	4	22	ARR-POST RF-AHC MALNUTRN	−
0.446	78	188	8	9	26	DMP SHOCK-CG RF-AHC APE	−
0.447	67	204	11	10	7	TRAUMA MVA FRACT	+
0.447	80	73	4	4	21	ARR-POST RF-CHC APE	+
0.447	77	14	1	0	19	F-AHC APE ATFIB	+
0.447	75	226	10	9	30	RF-AHC PNEUMONIA PUL-FIBROS	+

PRESS FOR MORE DETAILS

Match Weights

age (100)
daysicu (100)
ventdur (100)
chrdx1 (100)
ventnum (96)
vent (75)
addx1 (60)
centdur1 (59)
centnum (58)
apache (52)
artdur1 (70)
udur (65)
ucath (50)
artline (50)
artcom (50)
centline (25)
artnum (68)
addx2 (40)
addx3 (40)
dayscnc (33)
epidur (15)
daysflr (15)
sex (5)
ngtube (5)

Figure 6.5: Top ten matching cases in the NON-POSTOP group at the third phase.

Some interesting descriptive statistics about the presented case compared to the top ten matching cases appear when the "Summary Statistics" button is pressed. More information about the top ten matching patients can be obtained by pressing "+" or "-" which is close to "PRESS FOR MORE DETAILS". The "Continue" button is used to start another matching process.

Chapter 7

CONCLUSIONS

7.1 Summary

The thesis objectives have been achieved. Several case bases were reasonably constructed from an ICU medical database. New ideas on how to construct case bases from medical databases have been successfully explored. The case base reasoning shell, The Easy Reasoner, has been integrated with the existing user interface IDEAS for ICUs. A new version of IDEAS for ICUs (version 3.0) has been developed and tested.

Six different case bases were constructed based on Civetta's model and a detailed statistical analysis. One finding based on statistical analysis is that there is a significant difference between the POSTOP group and the NON-POSTOP group. A high correlation of 0.83 was found between DAYS_ICU and VENTHRS, for the entire ICU database. The interdependence of a new presented case and matched cases is better.

The indexing fields and field weights are decided based on the combination of (i) Civetta's [1993] model, (ii) statistical analysis, and (iii) the previous version of IDEAS for ICUs. This work adds scientific justification for the choice of weights and indexing fields.

Test results showed that the running speed has improved substantially using The Easy Reasoner (see section 6.6) instead of the ART-IM case-based reasoning tool. Based on

Taylor's thesis [1994], we know that, on average, more than 4 minutes were needed for the matching process using the ART-IM case-based reasoning tool, where around 50 fields were in the case base index and approximately 2000 patient cases were in the case base. Using The Easy Reasoner, under the same conditions, only approximately one minute is needed for performing the same process taking into account the time used to bring the output summary window up (a factor of approximately 4 times faster at least).

There is no end in scientific research, and things can be improved as long as we try to do so. During our research, we found some ideas very interesting but we didn't have time to pursue them. These ideas are stated in the next section as future work.

7.2 Future Work

More work can be done with ICD9_classes. We already know that, in the DECH ICU, the number of patients in some ICD9_classes is much larger than in others. Using case-based reasoning, for these ICD9_classes, we can create even smaller case bases (based on the ICD9_classes) than what we have now. It is very possible that the interdependence between a new case and matching cases will increase dramatically if we do so.

Several conclusions and recommendations can be made from observations of the ICU data set (according to the requirements of the physicians in the DECH ICU) using statistical analysis methods. The SAS program class.sas can be used to analyze the data. By modifying this SAS program and running it, one can get further information about the ICU data set and more detailed analysis of results.

It is true that The Easy Reasoner has proved to have a faster matching speed than ART-IM. However, compared with the ART-IM case-based reasoning tool, The Easy Reasoner is more like a black box. The final matching distance is the only value we can get from the matching process. To lessen the dependence on commercial tools over which researchers have no control, it is suggested that a case-based reasoning tool be designed and built from scratch. One could then experiment with different matching methodologies.

If we had our own case-based reasoning tool, we also could make the explanation part more attractive. It is very possible, after the top ten matching cases are retrieved based on a new presented case, that physicians would want to see the detailed contributions of each indexing field. These contributions should count for both the user supplied weight and the observed values of that field. We cannot do this using The Easy Reasoner because it only gives the final distance which combines the information from all fields. As a software developer, we must think ahead to give the end user more information and let them feel more comfortable with the results. From this point of view, this ability is considered very important.

Evaluation of IDEAS for ICUs version 3.0 is needed in the clinical setting of a hospital. Future work should include testing of concepts developed here in a pilot study of three to four weeks.

REFERENCES

[1] Beck, D. H., Taylor, B. L, Millar, B. & Smith, G. B., "Prediction of Outcome from Intensive Care: A Prospective Cohort Study Comparing Acute Physiology and Chronic Health Evaluation II and III Prognostic Systems in a United Kingdom Intensive Care Unit", *Critical Care Medicine*, pp. 9-15, 1997.

[2] Bradburn C., Zeleznikow J., Adams A., "FLORENCE: Synthesis of Case-Based and Model-Based Reasoning in a Nursing Care Planning System", *Computers in Nursing*, Vol. 11(1), pp. 20-24, 1993.

[3] Case Western Reserve University, "The CAse-based Menu Planner (CAMP)", *http://toros.ces.cwru.edu/~marling/camp.html*, 1997.

[4] Civetta, J. M., "Clinical Decision-Making", *Critical Care Medicine*, pp. 43-58, 1993.

[5] Civetta, J. M., "Prediction and Definition of Outcome", *Critical Care Medicine*, pp. 1873-1898, 1993.

[6] DECH, "Abbreviation for ICU Dbase (1988 through September 1992)", 1992.

[7] El-Gamal S., Rafeh M., Eissa I., "Case-Based Reasoning Algorithms Applied in a Medical Acquisition Tool", *Med. Inform.*, Vol. 18(2), pp. 149-162, 1993.

[8] Esserman, L., Jeffrey, B., and Lenert, L., "Potentially Ineffective Care -- a New Outcome to Assess the Limits of Critical Care", *JAMA*, Vol. 274(19), pp.1544-1511, 1995.

[9] Frize, M., Solven, F. G., Stevenson, M., Nickerson, B., Buskard, T., and Taylor, K., "Computer-Assisted Decision Support Systems for Patient Management in an Intensive Care Unit", Proceedings of the International Medical Informatics Association 8th World Congress on Medical Informatics (MEDINFO'95), pp.1009-1012, July 23-27, Vancouver, BC, 1995.

[10] Hicks, C. R., *Fundamental Concepts in the Design of Experiments*, Holt, Rinehart, and Winston, Inc, 1993.

[11] HSRG (Health Services Research Group), "ICD-9-CM International Classification of Diseases", *http://econ-www.newcastle.edu.au/hsrg/hypertexts/icd9cm.html*, 1995.

[12] Henderson, R. D., Deane, F. P., "User Expectations and Perceptions of a Patient Management Information System", *Computers in Nursing*, Vol. 14(3), pp. 188-193, 1996.

[13] Inference Corporation, "Case-Based Reasoning in ART-IM", Version 2.5, Inference Corporation, 550 North Continental Blvd. EL Segundo, California, 1991.

[14] Johnston, M. E., Langton, K. B. Haynes, R. B and Mathieu, A. "Effects of Computer-Based Clinical Decision Support System on Clinical Performance and Patient Outcome", *Annals of Internal Medicine*, pp. 135-142, 1994.

[15] Kahn, C. E., Longworth, N. J., Michalski, T. A., Pingree, M. J., "Intelligent Selection of Imaging Studies (ISIS)", *http://www.mcw.edu/midas/isis/*, Medical College of Wisconsin, 1997.

[16] Knaus W. A., Draper E. A., Wagner D. P., Zimmerman J. E., "An Evaluation of Outcome from Intensive Care in Major Medical Centers", *Annual of International Medicine*, Vol. 104, pp. 410-418, 1986.

[17] Koton P., "Reasoning about Evidence in Clinical Problem Solving", AAAI Spring Symposium Series, March 22-24, pp. 51-52, 1988.

[18] Krusinska, E., Babic, A., Chowdhury, S., Wigertz, o, Bodemar, G., Mathiesen, U., "Integrated Approach for Designing Medical Decision Support Systems with Knowledge Extracted from Clinical Databases by Statistical Methods", AMIA Inc. pp. 353-357, 1992.

[19] Lemeshow, S., Le Gall, J., "Modeling the Severity of Illness of ICU Patients", *JAMA*, Vol. 272 (13), pp.1049-1055, 1994.

[20] McGowan, H. C. E., "An Investigation of Method to Enhance the Performance of Artificial Neural Networks Used to Estimated ICU Outcomes", M. Sc. Eng. (EE) thesis, University of New Brunswick, Fredericton, N.B. Canada, 1996.

[21] Meade, M. O., Cook, D. J., "A Critical and Systematic Review of Illness Severity Scoring System in the Intensive Care Unit", *Current Science*, pp. 221-227, 1995.

[22] Mood, A. M., Graybill, F. A., Boes, D. C., *Introduction to the Theory of Statistics*, Mcgraw-Hill, Inc.,1974.

[23] Porter, B. W., Ray, B., Holte, R.C., "Concept Learning and Heuristic Classification in Weak Theory Domains", *Artificial Intelligence*, Vol. 45, pp. 229-263, 1990.

[24] Riesbeck, C. K, Shank, R. C., *Inside Case-Based Reasoning*, Lawrence Erlbaum Associates, Inc., Publishers, Mew Jersey, 1989.

[25] Salton G., "Developments in Automatic Text Retrieval", Science, Vol. 253, pp. 947-980, 1991.

[26] Santner, T. J., Duffy, D. E. *The Statistical Analysis of Discrete Data,* Spring-Verlag, New York, 1989.

[27] SAS Institute Inc., *SAS/STAT User's Guide, Version 6, Fourth Edition, Volumes 1&2,* SAS Institute Inc., Cary, NC, USA, 1990.

[28] Schefler, W. C., *Statistics for Health Professionals,* Addison-Wesley Publishing Company, Inc.,1984.

[29] Susan S., *Statistics for Health Professionals,* W.B.Saunders Company, Harcourt Brace Jovanovich, Inc., 1990.

[30] Taylor, K., "The Use of a Knowledge-Based System in the Management of Intensive Care Patients", M. Sc. Eng. (EE) thesis, University of New Brunswick, Fredericton, N.B. Canada, 1994.

[31] The Haley Enterprise, "The Easy Reasoner$^{\text{tm}}$", Eclipse 3.2g Reference Manual, pp.30-3 to 30-34, The Haley Enterprise, Inc., Sewickley, PA, USA, 1993.

[32] The Haley Enterprise, "Eclipse Help", Eclipse On-line Document, The Haley Enterprise, Inc., 1994.

[33] The Haley Enterprise, "Reasoning with Case Bases", Eclipse On-line Document, The Haley Enterprise, Inc.,1996.

[34] University of Chicago, "CBR Medical Demos and Project Descriptions", *http:///www.cs.uchicago.edu/cbr-med/html/demos.html,* 1996.

[35] Yu, V. L., Fagan, L. M., Bennett, S. W., Clancey, W. J., Scott, A. C. Hannigan, J. F., Blum, R. L., Buchanan, B. G. and Cohen, S. N. "Antimicrobial Selection by a Computer: a Blinded Evaluation by Infectious Disease Specialists", *Journal of the Americal Medical Assciation,* Vol. 242, No.12, pp.1279-1282, 1979.

[36] Vij, D., Babcock, R., Magilligan, D. J. "A Simplified Concept of Complete Physiological Monitoring of the Critical Ill Patient", *Heart & Lung,* Vol. 10, No.1, pp. 75-82, 1981.

[37] Wang, L. “Report on Preprocessing the DECH ICU Database” Medical IDEAS Research Group Technical Report TR-MIRG-97003, 7 pages, 1997.

Appendix I

ICU93.dbf Table Structure and Field Explanation

ICU93.dbf Table Structure and Field Explanation

by Lijuan Wang

July 22, 1996
(Revised on May 1, 1997)

Field	Field name	Type	Width	Index	Explanation
1	epidu_dur	numeric	2	n	epidual durations (days)
2	c_line_num	numeric	2	n	number of central line catheters
3	ccu	logical	1	n	from ccu overflow or not
4	em_surg	logical	1	n	from emergency surgery or not
5	lastname	character	10	n	patient's firstname
6	firstname	character	10	n	patient's lastname
7	age	numeric	2	n	patient's age
8	sex	character	1	n	patient's sex
9	h_num	character	6	n	hospital number
10	icu_num	character	3	n	icu number
11	teach	logical	1	n	teaching file or not
12	apache2	numeric	2	n	apache II score
13	ad_date	date	8	n	admission date
14	ad_time	numeric	2	n	admission time
15	ad_from	character	6	n	admitting from
16	ad_repeat	logical	1	n	repeat admission or not
17	ad_r_this	logical	1	n	this hospitalization or not
18	ad_r_prior	date	8	n	prior date
19	ad_md	character	10	n	admitting medical doctor
20	rf_md	character	10	n	repeat admitting medical doctor
21	ad_dept	character	4	n	admitting department
22	ad_dx1	character	10	n	admitting diagnosis no.1
23	ad_dx2	character	10	n	admitting diagnosis no.2
24	ad_dx3	character	10	n	admitting diagnosis no.3
25	ad_dx4	character	10	n	admitting diagnosis no.4
26	ad_dx5	character	10	n	admitting diagnosis no.5
27	ad_dx6	character	10	n	admitting diagnosis no.6
28	ad_dx7	character	10	n	admitting diagnosis no.7
29	ad_dx8	character	10	n	admitting diagnosis no.8
30	chr_dx1	character	10	n	chronic diagnosis no.1
31	chr_dx2	character	10	n	chronic diagnosis no.2
32	chr_dx3	character	10	n	chronic diagnosis no.3
33	chr_dx4	character	10	n	chronic diagnosis no.4

34	chr_dx5	character	10	n	chronic diagnosis no.5
35	chr_dx6	character	10	n	chronic diagnosis no.6
36	chr_dx7	character	10	n	chronic diagnosis no.7
37	chr_dx8	character	10	n	chronic diagnosis no.8
38	icu_dx1	character	10	n	icu diagnosis no.1
39	icu_dx2	character	10	n	icu diagnosis no.2
40	icu_dx3	character	10	n	icu diagnosis no.3
41	icu_dx4	character	10	n	icu diagnosis no.4
42	icu_dx5	character	10	n	icu diagnosis no.5
43	icu_dx6	character	10	n	icu diagnosis no.6
44	icu_dx7	character	10	n	icu diagnosis no.7
45	icu_dx8	character	10	n	icu diagnosis no.8
46	dis_date	date	8	n	discharging date
47	dis_time	numeric	2	n	discharging time
48	dis_to	character	6	n	discharging to
49	dis_condit	character	1	n	discharging condition
50	days_icu	numeric	3	n	days in icu
51	days_cnc	numeric	3	n	days in ccu
52	days_flr	numeric	3	n	days in floor
53	dis_md	character	10	n	discharging medical doctor
54	dis_dept	character	6	n	discharging department
55	dis_delay	numeric	2	n	delay between discharge order and implemention (days)
56	code_99_i	logical	1	n	code99 in icu or not
57	code_99_o	character	6	n	code99 in other location
58	autopsy	logical	1	n	autopsy or not
59	organ_req	logical	1	n	organ donation requested or not
60	organ_acc	logical	1	n	organ donation accepted or not
61	death	logical	1	n	death or not
62	ideathdate	date	8	n	death date in icu
63	ideathtime	numeric	2	n	death time in icu
64	ideathcaus	character	10	n	death cause in icu
65	fldeathdat	date	8	n	death date in floor
66	fldeathtim	numeric	2	n	death time in floor
67	fldeathcau	character	10	n	death cause in floor
68	odeathdate	date	8	n	death date in other location
69	odeathtime	numeric	2	n	death time in other location
70	odeathcaus	character	10	n	death cause in other location
71	otdeathtim	numeric	2	n	death time in other location (no data)
72	artlin	logical	1	n	arterial line catheter or not
73	artlin_num	numeric	2	n	number of arterial line catheters
74	artlindur1	numeric	2	n	arterial line catheter duration no.1 (days)
75	artlindur2	numeric	2	n	arterial line catheter duration no.2 (days)

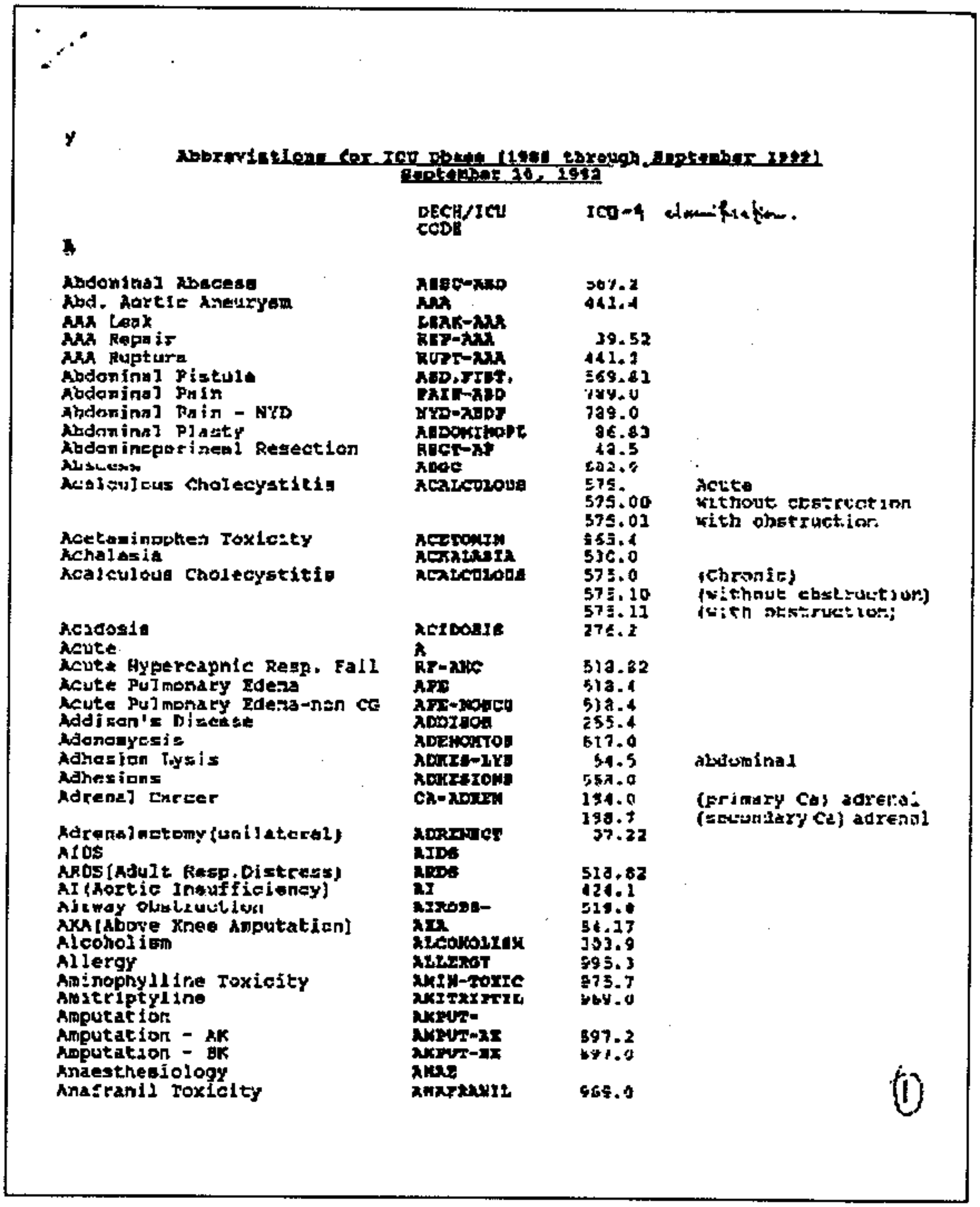
76	artlindur3	numeric	2	n	arterial line catheter duration no.3 (days)
77	artlin_com	logical	1	n	arterial line catheter complications or not
78	pacath	logical	1	n	pulmonary artery catheter or not
79	pacath_num	numeric	2	n	number of pulmonary artery catheters
80	pacathdur1	numeric	2	n	pulmonary artery catheter duration no.1 (days)
81	pacathdur2	numeric	2	n	pulmonary artery catheter duration no.2 (days)
82	pacathdur3	numeric	2	n	pulmonary artery catheter duration no.3 (days)
83	pacath_com	logical	1	n	pulmonary artery catheter complications or not
84	c_line	logical	1	n	central line catheter or not
85	c_linedur1	numeric	2	n	central line catheter duration no.1 (days)
86	c_linedur2	numeric	2	n	central line catheter duration no.2 (days)
87	c_linedur3	numeric	2	n	central line catheter duration no.3 (days)
88	c_line_com	logical	1	n	central line catheter complications or not
89	vent	logical	1	n	ventilation or not
90	vent_num	numeric	2	n	number of times of ventilated
91	venthrs	numeric	4	n	number of hours of ventilated
92	vent_com	logical	1	n	ventilation complications or not
93	chstub	logical	1	n	chest tube or not
94	chstubnum	numeric	2	n	total number of chest tubes
95	chstubdur1	numeric	2	n	chest tube duration no. 1 (days)
96	chstubdur2	numeric	2	n	chest tube duration no. 2 (days)
97	chstubdur3	numeric	2	n	chest tube duration no. 3 (days)
98	chstub_com	logical	1	n	chest tube complications or not
99	u_cath	logical	1	n	urine catheter or not
100	u_cath_dur	numeric	3	n	urine catheter durations (days)
101	u_cath_com	logical	1	n	urine catheter complications or not
102	tpn	logical	1	n	total pureutesal nutrition
103	tpn_com	logical	1	n	total pureutesal nutrition complications or not
104	ng_tub	logical	1	n	nasogastric tube or not
105	epidur	logical	1	n	epidural or not
106	epidur_com	logical	1	n	epidural complications or not
107	o_procedur	memo	10	n	other procedure
108	inft	logical	1	n	infection or not
109	inft_ad	logical	1	n	infection on admission or not

110	inft_ad_1	character	10	n	infection on admission diagnosis no.1
111	inft_ad_2	character	10	n	infection on admission diagnosis no.2
112	inft_icu	logical	1	n	infection acquired in icu or not
113	inft_icu_1	character	10	n	infection acquired in icu diagnosis no.1
114	inft_icu_2	character	10	n	infection acquired in icu diagnosis no.2
115	inft_icu_3	character	10	n	infection acquired in icu diagnosis no.3
116	comentproc	memo	10	n	if more than three procedures or complications or infection

Appendix II

The Copy of the First Page of "Abbreviation for ICU Dbase (1998 through September 1992)"

The Copy of the First Page of "Abbrevation for ICU Dbase (1998 through September 1992)"



Abbreviations for ICU Dbase (1988 through September 1992)
September 10, 1992

A

	DECH/ICU CODE	ICD-9	classification.
Abdominal Abscess	ABSC-ABD	567.2	
Abd. Aortic Aneuryem	AAA	441.4	
AAA Leak	LEAK-AAA		
AAA Repair	REP-AAA	39.52	
AAA Rupture	RUPT-AAA	441.3	
Abdominal Fistula	ABD.FIST.	569.81	
Abdominal Pain	PAIN-ABD	789.0	
Abdominal Pain - NYD	NYD-ABDP	789.0	
Abdominal Plasty	ABDOMINOPL	86.83	
Abdominoperineal Resection	RECT-AP	48.5	
Abscess	ABSC	682.9	
Acalculous Cholecystitis	ACALCULOUS	575.	Acute
		575.00	without obstruction
		575.01	with obstruction
Acetaminophen Toxicity	ACETOMIN	965.4	
Achalasia	ACHALASIA	530.0	
Acalculous Cholecystitis	ACALCULOUS	575.0	(Chronic)
		575.10	(without obstruction)
		575.11	(with obstruction)
Acidosis	ACIDOSIS	276.2	
Acute	A		
Acute Hypercapnic Resp. Fail	RF-AHC	518.82	
Acute Pulmonary Edema	APE	518.4	
Acute Pulmonary Edema-non CG	APE-NONCG	518.4	
Addison's Disease	ADDISON	255.4	
Adenomyosis	ADENOMYOS	617.0	
Adhesion Lysis	ADHES-LYS	54.5	abdominal
Adhesions	ADHESIONS	568.0	
Adrenal Cancer	CA-ADREN	194.0	(primary Ca) adrenal
		198.7	(secundary Ca) adrenal
Adrenalectomy(unilateral)	ADRENECT	07.22	
AIDS	AIDS		
ARDS(Adult Resp.Distress)	ARDS	518.82	
AI(Aortic Insufficiency)	AI	424.1	
Airway Obstruction	AIROBS-	519.8	
AKA(Above Knee Amputation)	AKA	84.17	
Alcoholism	ALCOHOLISM	303.9	
Allergy	ALLERGY	995.3	
Aminophylline Toxicity	AMIN-TOXIC	975.7	
Amitriptyline	AMITRIPTYL	969.0	
Amputation	AMPUT-		
Amputation - AK	AMPUT-AK	897.2	
Amputation - BK	AMPUT-BK	897.0	
Anaesthesiology	ANAE		
Anafranil Toxicity	ANAFRANIL	969.0	

1

Appendix III

This Appendix contains the SAS Program class.sas used to perform analysis of the ICU93.dbf data. These data were broken into seven comma delimited data files labeled data1.txt, data2.txt, data3.txt, data4.txt, data5.txt, data6.txt, data7.txt. This splitting into seven files was necessary due to the large size of the database. When running this SAS program, you should comment out most parts of the "STATISTICAL ANALYSIS STEP", or the program will generate many thousands of pages of output.

For example, all of the operations are commented out in the Appendix as shown, expect for the last five operations, starting with "To test the relation between SEX and DEATH".

The file postop.txt is described in Chapter 3. The file index.txt is simply an auxiliary file to assist with plotting and analysis.

```
/*************************************************************************************/
/* Program Name : class.sas                                                          */
/* Programmer Name: Lijuan Wang                                                      */
/* Last Revised Date : Nov. 18, 1997                                                 */
/*                                                                                   */
/* Note: (1) This program is used to read external ICU data files and then           */
/*           proceed some statistical analysis;                                      */
/*       (2) To avoid overwhelming output, it is highly recommended to run the       */
/*           program partly.                                                         */
/*                                                                                   */
/*************************************************************************************/


              /**************************************************************/
              /*                                                            */
              /*                        DATA STEP                           */
              /*                                                            */
              /*  To create SAS data set 'icuclass' based on icu data sets  */
              /*                                                            */
              /**************************************************************/


/*************************************************************************************/
/*                                                                                   */
/*  To read eternal files 'f:/index.txt' and 'f:/postop.txt'                         */
/*                                                                                   */
/*************************************************************************************/

data icuclass;

/*infile 'f:/index.txt';
input count @@;*/

infile 'f:/postop.txt'  DELIMITER = ',';
input postop @@;


/*************************************************************************************/
/*                                                                                   */
/*  To read eternal file 'f:/data1.txt' and use some other commands to process data */
/*                                                                                   */
/*************************************************************************************/

infile 'f:/data1.txt' DELIMITER = ',';
length ad_md $12;
length rf_md $12;
length ad_dx1 $12;
input epidu_du clinenum ccu em_surg age sex $ h_num $ icu_num $ teach apache2 ad_date $
      ad_time $ ad_from $ adrepeat adrthis adrprio $ ad_md $ rf_md $ ad_dept $ ad_dx1 $
      icd9d1 classd1 @@;

if epidu_du = -1 then Repidu_d = .;
else Repidu_d = epidu_du;

if clinenum = -1 then Rclinenu = .;
else Rclinenu = clinenum;

if age = -1 then Rage = .;
else Rage = age;

if apache2 = -1 then Rapache2 = .;
else if  apache2 > 70 then Rapache2 = .;
else Rapache2 = apache2;


/*************************************************************************************/
/*                                                                                   */
/*  To read eternal file 'f:/data2.txt' and use some other commands to process data */
/*                                                                                   */
/*************************************************************************************/

infile 'f:/data2.txt' DELIMITER = ',';
length ad_dx2 $ 12;
length ad_dx3 $ 12;
length ad_dx4 $ 12;
length ad_dx5 $ 12;
length ad_dx6 $ 12;
length ad_dx7 $ 12;
length ad_dx8 $ 12;
input ad_dx2 $ icd9d2 classd2 ad_dx3 $ ad_dx4 $ ad_dx5 $ ad_dx6 $ ad_dx7 $ ad_dx8 $@@;


/************************************************************************************/
/*                                                                                  */
/*  To read eternal file 'f:/data3.txt' and use some other commands to process data */
/*                                                                                  */
/************************************************************************************/

infile 'f:/data3.txt' DELIMITER = ',';
length chr_dx1 $ 12;
length chr_dx2 $ 12;
length chr_dx3 $ 12;
length chr_dx4 $ 12;
length chr_dx5 $ 12;
length chr_dx6 $ 12;
length chr_dx7 $ 12;
length chr_dx8 $ 12;
length icu_dx1 $ 12;
length icu_dx2 $ 12;
length icu_dx3 $ 12;
length icu_dx4 $ 12;
length icu_dx5 $ 12;
length icu_dx6 $ 12;
length icu_dx7 $ 12;
length icu_dx8 $ 12;
input chr_dx1 $ icd9c1 classc1 chr_dx2 $ chr_dx3 $ chr_dx4 $ chr_dx5 $ chr_dx6 $ chr_dx7 $
chr_dx8 $
icu_dx1 $ icu_dx2 $ icu_dx3 $ icu_dx4  $ icu_dx5 $ icu_dx6 $ icu_dx7 $ icu_dx8 $@@;


/************************************************************************************/
/*                                                                                  */
/*  To read eternal file 'f:/data4.txt' and use some other commands to process data */
/*                                                                                  */
/************************************************************************************/

infile 'f:/data4.txt' DELIMITER = ',';
length dis_date $12;
length dis_md $12;
input dis_date $ dis_time dis_to $ dis_cond $ days_icu days_ccu days_flr dis_md $
      dis_dept $ dis_dela cod_99_i cod_99_o $ autopcy organ_re organ_ac @@;

if days_icu = -1 then Rdays_ic = .;
else if days_icu > 100 then Rdays_ic = .;
else Rdays_ic = days_icu;

if days_ccu = -1 then Rdays_cc = .;
else Rdays_cc = days_ccu;

if days_flr = -1 then Rdays_fl = .;
else Rdays_fl = days_flr;

if dis_dela = -1 then Rdis_del = .;
else Rdis_del = dis_dela;


/************************************************************************************/
/*                                                                                  */
/*  To read eternal file 'f:/data5.txt' and use some other commands to process data */
/*                                                                                  */
/************************************************************************************/


infile 'f:/data5.txt' DELIMITER = ',';
length ideathca $ 12;
length fldeathc $ 12;
length odeathca $ 12;
input death ideathda $ ideathti ideathca $ fldeathd $ fldeatht fldeathc $ odeathda $
```

```
     odeathti odeathca $ otdeatht artlin artlin_n artlind1 artlind2 artlind3 artlinco @@;
if ideathti = -1 then Rideatht = .;
else Rideatht = ideathti;

if fldeatht = -1 then Rfldeatt = .;
else Rfldeatt = fldeatht;


/*********************************************************************************/
/*                                                                               */
/*  To read eternal file 'f:/data6.txt' and use some other commands to process data */
/*                                                                               */
/*********************************************************************************/


infile 'f:/data6.txt' DELIMITER = ',';
input pacath pacath_n pacathd1 pacathd2 pacathd3 pacathco cline clined1 clined2 clined3
clinecom vent ventnum venthrs ventcom chstub chstubn chstubd1 chstubd2 chstubd3 chstubco
@@;

if artlin_n = -1 then Rartli_n = .;
else Rartli_n = artlin_n;

if artlind1 = -1 then Rartlid1 = .;
else if artlind1 > 40 then Rartlid1 = .;
else Rartlid1 = artlind1;

if artlind2 = -1 then Rartlid2 = .;
else Rartlid2 = artlind2;

if artlind3 = -1 then Rartlid3 = .;
else Rartlid3 = artlind3;

if pacath_n = -1 then Rpacat_n = .;
else Rpacat_n = pacath_n;

if pacathd1 = -1 then Rpacatd1 = .;
else Rpacatd1 = pacathd1;

if pacathd2 = -1 then Rpacatd2 = .;
else Rpacatd2 = pacathd2;

if pacathd3 = -1 then Rpacatd3 = .;
else Rpacatd3 = pacathd3;

if clined1 = -1 then Rclined1 = .;
else Rclined1 = clined1;

if clined2 = -1 then Rclined2 = .;
else Rclined2 = clined2;

if clined3 = -1 then Rclined3 = .;
else Rclined3 = clined3;

if ventnum = -1 then Rventnum = .;
else Rventnum = ventnum;

if venthrs = -1 then Rventhrs = .;
else Rventhrs = venthrs;

if chstubn = -1 then Rchstubn = .;
else Rchstubn = chstubn;

if chstubd1 = -1 then Rchstud1 = .;
else Rchstud1 = chstubd1;

if chstubd2 = -1 then Rchstud2 = .;
else Rchstud2 = chstubd2;

/*********************************************************************************/
/*                                                                               */
/*  To read eternal file 'f:/data7.txt' and use some other commands to process data */
/*                                                                               */
/*********************************************************************************/
```

```
infile 'f:/data7.txt' DELIMITER = ',';
length oprocedu $ 12;
length inftad1 $ 12;
length inftad2 $ 12;
length infticu1 $12;
length infticu2 $12;
length infticu3 $ 12;
length comentpr $ 12;
input ucath ucathdur ucathcom tpn tpncom ngtub epidur epdurcom oprocedu $
      inft inftad inftad1 $ inftad2 $ infticu infticu1 $ infticu2 $ infticu3 $ comentpr $
@@;

if ucathdur = -1 then Rucathdu = .;
else if ucathdur > 100 then Rucathdu = .;
else Rucathdu = ucathdur;
run;


              /*****************************************************************/
              /*                                                               */
              /*                STATISTICAL ANALYSIS STEP                      */
              /*                                                               */
              /* To obtain histogram, Pie Chart, descriptive statistics and    */
              /* Pearson correlation coeffients, and to test the relationship */
              /* either between a numerical variable and a nominal variable    */
              /* or between two nominal variable.based on SAS data set         */
              /* 'icuclass'                                                    */
              /*                                                               */
              /*                                                               */
              /*****************************************************************/


/************************************************************************************/
/*                                                                                  */
/*  To obtain frequency counts, pie chart and histogram                             */
/*                                                                                  */
/************************************************************************************/

/*proc chart data = icuclass; */
  /*hbar epidu_du clinenum ccu em_surg age sex teach apache2
       ad_from adrepeat adrthis ad_dept;*/

  /*hbar ad_dx1 ad_dx2 ad_dx3 ad_dx4 ad_dx5 ad_dx6 ad_dx7 ad_dx8;
      chr_dx1 chr_dx2 chr_dx3 chr_dx4 chr_dx5 chr_dx6 chr_dx7 chr_dx8
      icu_dx1 icu_dx2 icu_dx3 icu_dx4 icu_dx5 icu_dx6 icu_dx7 icu_dx8;*/

  /*hbar dis_to  dis_cond  days_icu days_ccu days_flr
      dis_dept dis_dela cod_99_i cod_99_o autopcy organ_re organ_ac
      death ideathti ideathca fldeatht fldeathc odeathda
      odeathti odeathca otdeatht artlin artlin_n artlind1 artlind2 artlind3 artlinco;*/

  /*hbar pacath pacath_n pacathd1 pacathd2 pacathd3 pacathco cline clined1 clined2 clined3
      clinecom vent ventnum venthrs ventcom chstub chstubn chstubd1 chstubd2 chstubd3
      chstubco;*/

  /*hbar ucath ucathdur ucathcom tpn tpncom ngtub epidur epdurcom oprocedu
      inft inftad inftad1 inftad2 infticu infticu1 infticu2 infticu3 comentpr;*/


/*proc chart data = icuclass;
   pie classd1/midpoints = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18;
   pie postop/midpoints = 0 1;
*/

/*proc chart;
   hbar Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
*/
```

```
/************************************************************************************/
/*                                                                                  */
/*   To print some data                                                             */
/*                                                                                  */
/************************************************************************************/

/*proc print;
   var Repidu_d Rclinenu Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
*/

/************************************************************************************/
/*                                                                                  */
/*   To obtain some descriptive statistics, including min, max, median etc          */
/*                                                                                  */
/************************************************************************************/

/*proc univariate freq normal plot;
   var epidu_du clinenum age apache2
       days_icu days_ccu days_flr dis_dela
       ideathti fldeatht
       artlin_n artlind1 artlind2 artlind3 pacath_n pacathd1 pacathd2 pacathd3
       clined1 clined2 clined3 ventnum venthrs chstubn chstubd1 chstubd2
       ucathdur; */


/*proc univariate freq normal plot;
  var Repidu_d Rclinenu Rage Rapache2
      Rdays_ic Rdays_cc Rdays_fl Rdis_del
      Rideatht Rfldeatt
      Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
      Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
      Rucathdu;
*/

/************************************************************************************/
/*                                                                                  */
/*   To obtain some correlation coefficients for whole icu data set                 */
/*                                                                                  */
/************************************************************************************/

/*proc corr;
   var epidu_du clinenum age apache2
       days_icu days_ccu days_flr dis_dela
       ideathti fldeatht
       artlin_n artlind1 artlind2 artlind3 pacath_n pacathd1 pacathd2 pacathd3
       clined1 clined2 clined3 ventnum venthrs chstubn chstubd1 chstubd2
       ucathdur;
*/

/*proc corr;
  var Repidu_d Rclinenu Rage Rapache2
      Rdays_ic Rdays_cc Rdays_fl Rdis_del
      Rideatht Rfldeatt
      Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
      Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
      Rucathdu;
      with Rage;
*/

/************************************************************************************/
/*                                                                                  */
/*   To obtain correlation coefficients grouped by postop                           */
/*                                                                                  */
/************************************************************************************/

/*proc sort;
   by postop;  */

/*proc corr;
   by postop;
   var epidu_du clinenum age apache2 days_ccu days_flr dis_dela ideathti fldeatht
       artlin_n artlind1 artlind2 artlind3 pacath_n pacathd1 pacathd2 pacathd3
       clined1 clined2 clined3 ventnum venthrs chstubn chstubd1 chstubd2 ucathdur;
   with days_icu;

proc corr;
   by postop;
   var epidu_du clinenum age apache2 days_icu days_ccu days_flr dis_dela ideathti fldeatht
       artlin_n artlind1 artlind2 artlind3 pacath_n pacathd1 pacathd2 pacathd3
       clined1 clined2 clined3 ventnum chstubn chstubd1 chstubd2 ucathdur;
   with venthrs;

proc corr spearman data = icuclass outs = total1;
   var Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
   with Rdays_ic;
proc print data = total1;

proc corr spearman data = icuclass outs = total2;
   var Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
   with Rventhrs;
proc print data = total2;

*/
/*proc sort data = icuclass;
   by postop;
proc corr spearman data = icuclass  outs = post1;
   by postop;
   var Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
   with Rdays_ic;
proc print data = post1;


proc corr spearman data = icuclass outs = post2;
   by postop;
   var Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
   with Rventhrs;
proc print data = post2;


proc sort data = icuclass;
   by postop;
proc corr spearman data = icuclass  outs = post3;
   by postop;
   var Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
   with Rage;
proc print data = post3;


proc sort data = icuclass;
   by postop;
```

```
proc corr spearman data = icuclass  outs = post4;
   by postop;
   var Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
   with Rapache2;
proc print data = post4;


proc sort data = icuclass;
   by postop;
proc corr spearman data = icuclass  outs = post5;
   by postop;
   var Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
   with Rartlid1;
proc print data = post5;
*/

/*********************************************************************************/
/*                                                                               */
/*   To obtain correlation coefficients grouped by class                         */
/*                                                                               */
/*********************************************************************************/

/*proc sort;
by class;

proc corr;
   by class;
   var epidu_du clinenum age apache2  days_ccu days_flr dis_dela ideathti fldeatht
odeathti
       artlin artlin_n artlind1 artlind2 artlind3 pacath_n pacathd1 pacathd2 pacathd3
       clined1 clined2 clined3 ventnum chstubn chstubd1 chstubd2 chstubd3 ucathdur;
   with days_icu;

proc corr;
   by class;
   var epidu_du clinenum age apache2 days_icu days_ccu days_flr dis_dela ideathti fldeatht
       odeathti artlin artlin_n artlind1 artlind2 artlind3 pacath_n pacathd1 pacathd2
pacathd3
       clined1 clined2 clined3 ventnum chstubn chstubd1 chstubd2 chstubd3 ucathdur;
   with venthrs;
*/

/*********************************************************************************/
/*                                                                               */
/*   To test the relationship between numerical variables and death (incl. postop)   */
/*                                                                               */
/*********************************************************************************/

/*proc sort data = icuclass;
   by postop;
 */

/*proc means data = icuclass;
   var death;
   by postop;


proc freq data = icuclass;
   tables postop*death/ chisq measures;
*/

/*proc glm data = icuclass;
   class death;
   model Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu = death;
   means death / SNK;
 */

  /* lsmeans death; */
  /* by postop;     */

/*proc sort data = icuclass;
  by death;

proc univariate data = icuclass;
   var Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
   by death;
*/


/*************************************************************************************/
/*                                                                                   */
/*   To test the relationship between postop and sex                                */
/*                                                                                   */
/*************************************************************************************/

/*proc sort data = icuclass;
   by postop;


proc freq data = icuclass;
   tables postop*sex/ chisq measures;
*/

/*************************************************************************************/
/*                                                                                   */
/*   To test the relationship between numerical variables and sex (incl. postop)    */
/*                                                                                   */
/*************************************************************************************/

/*proc glm data = icuclass;
   class sex;
   model Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu = sex;
   means sex /SNK;
*/
  /* lsmeans sex; */
  /* by postop;   */

/*proc sort data = icuclass;
  by sex;

proc univariate data = icuclass;
   var Repidu_d Rclinenu Rage Rapache2
       Rdays_ic Rdays_cc Rdays_fl Rdis_del
       Rideatht Rfldeatt
       Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
       Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
       Rucathdu;
   by sex;
*/

/*************************************************************************************/
/*                                                                                   */
/* To test the relationship between numerical variables and artlinco                */
/*                                                                                   */
/*************************************************************************************/
```

```
/*proc sort data = icuclass;
   by postop;


proc glm data = icuclass;
   class artlinco;
   model Repidu_d Rclinenu Rage Rapache2
        Rdays_ic Rdays_cc Rdays_fl Rdis_del
        Rideatht Rfldeatt
        Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
        Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
        Rucathdu = artlinco;
   means artlinco /SNK;
   by postop;

proc sort data = icuclass;
  by artlinco;

proc univariate data = icuclass;
   var Repidu_d Rclinenu Rage Rapache2
        Rdays_ic Rdays_cc Rdays_fl Rdis_del
        Rideatht Rfldeatt
        Rartli_n Rartlid1 Rartlid2 Rartlid3 Rpacat_n Rpacatd1 Rpacatd2 Rpacatd3
        Rclined1 Rclined2 Rclined3 Rventnum Rventhrs Rchstubn Rchstud1 Rchstud2
        Rucathdu;
   by artlinco;
*/

/*****************************************************************************/
/*                                                                           */
/*   To test the relationship between postop and artlinco                    */
/*                                                                           */
/*****************************************************************************/
/*
proc freq data = icuclass;
   tables postop*artlinco/ chisq measures;
*/
/*****************************************************************************/
/*                                                                           */
/*   To test the relationship between postop and ngtub                       */
/*                                                                           */
/*****************************************************************************/
/*
proc freq data = icuclass;
   tables postop*ngtub/ chisq measures;
*/
/*****************************************************************************/
/*                                                                           */
/*   To test the relationship between postop and epidur                      */
/*                                                                           */
/*****************************************************************************/
/*
proc freq data = icuclass;
   tables postop*epidur/ chisq measures;
*/
/*****************************************************************************/
/*                                                                           */
/*   To test the relationship between ngtub and death                        */
/*                                                                           */
/*****************************************************************************/
/*
proc freq data = icuclass;
   tables ngtub*death/ chisq measures;
*/
/*****************************************************************************/
/*                                                                           */
/*   To test the relationship between death and epidur                       */
/*                                                                           */
/*****************************************************************************/
/*
proc freq data = icuclass;
   tables epidur*death/ chisq measures;
*/
```

```
/*******************************************************************************/
/*                                                                             */
/*   To test the relationship between artlinco and death                       */
/*                                                                             */
/*******************************************************************************/
/*
proc freq data = icuclass;
   tables artlinco*death/ chisq measures;
*/
/*******************************************************************************/
/*                                                                             */
/*   To test the relationship between sex and death                            */
/*                                                                             */
/*******************************************************************************/

proc freq data = icuclass;
   tables sex*death/ chisq measures;

/*******************************************************************************/
/*                                                                             */
/*   To test the relationship between sex and artlinco                         */
/*                                                                             */
/*******************************************************************************/

proc freq data = icuclass;
   tables sex*artlinco/ chisq measures;

/*******************************************************************************/
/*                                                                             */
/*   To test the relationship between sex and epidur                           */
/*                                                                             */
/*******************************************************************************/

proc freq data = icuclass;
   tables sex*epidur/ chisq measures;

/*******************************************************************************/
/*                                                                             */
/*   To test the relationship between sex and ngtub                            */
/*                                                                             */
/*******************************************************************************/

proc freq data = icuclass;
   tables sex*ngtub/ chisq measures;

/*******************************************************************************/
/*                                                                             */
/*   To test the relationship between sex and postop                           */
/*                                                                             */
/*******************************************************************************/

proc freq data = icuclass;
   tables sex*postop/ chisq measures;

run;
```

Appendix IV

New PATIENT.dbf and Old PATIENT.dbf Table Structures

New PATIENT.dbf Table Structure

FIELD_NAME	FIELD_TYPE	FIELD_LEN	FIELD_DEC	FIELD_IDX
KEYVALUE	N	10	0	N
SEX	C	1	0	N
EPIDUR	N	4	0	N
CENTNUM	N	4	0	N
CCCOFLOW	C	10	0	N
CCU	C	1	0	N
EM_SURG	C	1	0	N
LASTNAME	C	40	0	N
FIRSTNAME	C	20	0	N
AGE	N	3	0	N
HOSPNUM	C	15	0	N
ICUNUM	C	15	0	N
TEACH	C	10	0	N
APACHE	N	4	0	N
ADDATE	D	8	0	N
ADTIME	N	4	0	N
ADFROM	C	15	0	N
ADREPEAT	C	1	0	N
ADREPHOS	C	1	0	N
ADREPPRIOR	D	8	0	N
ADMD	C	20	0	N
RFMD	C	10	0	N
ADDEPT	C	15	0	N
ADDX1	C	10	0	N
ADDX2	C	10	0	N
ADDX3	C	10	0	N
ADDX4	C	10	0	N
ADDX5	C	10	0	N
ADDX6	C	10	0	N
ADDX7	C	10	0	N
ADDX8	C	10	0	N
CHRDX1	C	10	0	N
CHRDX2	C	10	0	N
CHRDX3	C	10	0	N
CHRDX4	C	10	0	N
CHRDX5	C	10	0	N
CHRDX6	C	10	0	N
CHRDX7	C	10	0	N
CHRDX8	C	10	0	N
ICUEVENT1	C	10	0	N
ICUEVENT2	C	10	0	N
ICUEVENT3	C	10	0	N
ICUEVENT4	C	10	0	N
ICUEVENT5	C	10	0	N
ICUEVENT6	C	10	0	N
ICUEVENT7	C	10	0	N
ICUEVENT8	C	10	0	N
DISDATE	D	8	0	N
DISTIME	N	4	0	N
DISTO	C	15	0	N
DISCOND	C	10	0	N
DAYSICU	N	4	0	N
DAYSCNC	N	4	0	N
DAYSFLR	N	4	0	N
DAYSTOT	N	4	0	N
DISMD	C	20	0	N
DISDEPT	C	15	0	N
DISDELAY	N	4	0	N
CODE99I	C	10	0	N
CODE99O	C	10	0	N
AUTOPSY	C	1	0	N
ORGANREQ	C	1	0	N
ORGANACC	C	1	0	N
DEATH	C	5	0	N
DDI_DATE	D	8	0	N
DDI_TIME	N	4	0	N
DDI_CAUSE	C	10	0	N
DDF_DATE	D	8	0	N
DDF_TIME	N	4	0	N

DDF_CAUSE	C	10	0	N
DDO_DATE	D	8	0	N
DDO_TIME	N	4	0	N
DDO_CAUSE	C	10	0	N
PERIPHIV	C	1	0	N
PERIPHDUR	N	4	0	N
PERIPHCOM	C	1	0	N
ARTLINE	C	1	0	N
ARTNUM	N	4	0	N
ARTDUR1	N	4	0	N
ARTDUR2	N	4	0	N
ARTDUR3	N	4	0	N
ARTCOM	C	1	0	N
PACATH	C	1	0	N
PANUM	N	4	0	N
PADUR1	N	4	0	N
PADUR2	N	4	0	N
PADUR3	N	4	0	N
PACOM	C	1	0	N
CENTLINE	C	1	0	N
CENTDUR1	N	4	0	N
CENTDUR2	N	4	0	N
CENTDUR3	N	4	0	N
CENTCOM	C	1	0	N
VENT	C	1	0	N
VENTNUM	N	4	0	N
VENTDUR	N	4	0	N
VENTCOM	C	1	0	N
CHSTTUBE	C	1	0	N
CHSTNUM	N	4	0	N
CHSTDUR1	N	4	0	N
CHSTDUR2	N	4	0	N
CHSTDUR3	N	4	0	N
CHSTCOM	C	1	0	N
UCATH	C	1	0	N
UDUR	N	4	0	N
UCOM	C	1	0	N
TPN	C	1	0	N
TPNCOM	C	1	0	N
NGTUBE	C	1	0	N
EPIDURAL	C	1	0	N
EPICOM	C	1	0	N
O_PROCEDUR	M	10	0	N
INFECTION	C	1	0	N
INFAD1	C	1	0	N
INFAD2	C	10	0	N
INTAD3	C	10	0	N
INFICU1	C	1	0	N
INFICU2	C	10	0	N
INFICU3	C	10	0	N
INFICU4	C	10	0	N
AUTOPSYRE	M	10	0	N
HGT_CM	N	10	0	N
WGT_KG	N	10	0	N
BSA	N	10	0	N
MED1	C	10	0	N
MED2	C	10	0	N
MED3	C	10	0	N
MED4	C	10	0	N
MED5	C	10	0	N
MED6	C	10	0	N
MED7	C	10	0	N
MED8	C	10	0	N
COMENTPROC	M	10	0	N
STUDYREP	M	10	0	N

Old PATIENT.dbf Table Structure

FIELD_NAME	FIELD_TYPE	FIELD_LEN	FIELD_DEC	FIELD_IDX
KEYVALUE	N	4	0	N
SEX	C	1	0	N
EPIDUR	N	4	0	N
CENTNUM	N	4	0	N
CCCOFLOW	C	10	0	N
LASTNAME	C	40	0	N
FIRSTNAME	C	20	0	N
AGE	N	3	0	N
HOSPNUM	C	15	0	N
ICUNUM	C	15	0	N
TEACH	C	10	0	N
APACHE	N	4	0	N
ADDATE	D	8	0	N
ADTIME	N	4	0	N
ADFROM	C	15	0	N
ADREPEAT	C	1	0	N
ADREPHOS	C	1	0	N
ADREPPRIOR	C	8	0	N
ADMD	C	20	0	N
ADDEPT	C	15	0	N
ADDX1	C	70	0	N
ADDX2	C	70	0	N
ADDX3	C	70	0	N
ADDX4	C	70	0	N
ADDX5	C	70	0	N
ADDX6	C	70	0	N
ADDX7	C	70	0	N
ADDX8	C	10	0	N
DISDATE	D	8	0	N
DISTIME	N	4	0	N
DISTO	C	15	0	N
DISCOND	C	10	0	N
DAYSICU	N	4	0	N
DAYSCNC	N	4	0	N
DAYSFLR	N	4	0	N
DAYSTOT	N	4	0	N
DISMD	C	20	0	N
DISDEPT	C	15	0	N
DISDELAY	N	4	0	N
CODE99I	C	10	0	N
CODE99O	C	10	0	N
AUTOPSY	C	1	0	N
ORGANREQ	C	1	0	N
ORGANACC	C	1	0	N
DEATH	C	5	0	N
DDI_DATE	D	8	0	N
DDI_TIME	N	4	0	N
DDI_CAUSE	C	70	0	N
DDF_DATE	D	8	0	N
DDF_TIME	N	4	0	N
DDF_CAUSE	C	10	0	N
DDO_DATE	D	8	0	N
DDO_TIME	N	4	0	N
DDO_CAUSE	C	10	0	N
PERIPHIV	C	1	0	N
PERIPHDUR	N	4	0	N
PERIPHCOM	C	1	0	N
ARTLINE	C	1	0	N
ARTNUM	N	4	0	N
ARTDUR1	N	4	0	N
ARTDUR2	N	4	0	N
ARTDUR3	N	4	0	N
ARTCOM	C	1	0	N
PACATH	C	1	0	N
PANUM	N	4	0	N
PADUR1	N	4	0	N
PADUR2	N	4	0	N
PADUR3	N	4	0	N
PACOM	C	1	0	N

CENTLINE	C	1	0	N
CENTDUR1	N	4	0	N
CENTDUR2	N	4	0	N
CENTDUR3	N	4	0	N
CENTCOM	C	1	0	N
VENT	C	1	0	N
VENTNUM	N	4	0	N
VENTDUR	N	4	0	N
VENTCOM	C	1	0	N
CHSTTUBE	C	1	0	N
CHSTNUM	N	4	0	N
CHSTDUR1	N	4	0	N
CHSTDUR2	N	4	0	N
CHSTDUR3	N	4	0	N
CHSTCOM	C	1	0	N
UCATH	C	1	0	N
UDUR	N	4	0	N
UCOM	C	1	0	N
TPN	C	1	0	N
TPNCOM	C	1	0	N
NGTUBE	C	1	0	N
EPIDURAL	C	1	0	N
EPICOM	C	1	0	N
INFECTION	C	1	0	N
INFAD1	C	1	0	N
INFAD2	C	1	0	N
INFAD3	C	1	0	N
INFICU1	C	1	0	N
INFICU2	C	1	0	N
INFICU3	C	1	0	N
INFICU4	C	1	0	N
AUTOPSYREP	M	10	0	N
HGT_CM	N	10	2	N
WGT_KG	N	10	2	N
BSA	N	10	2	N
ICUEVENT1	C	70	0	N
ICUEVENT2	C	70	0	N
ICUEVENT3	C	70	0	N
ICUEVENT4	C	70	0	N
ICUEVENT5	C	70	0	N
ICUEVENT6	C	70	0	N
ICUEVENT7	C	70	0	N
ICUEVENT8	C	10	0	N
MED1	C	70	0	N
MED2	C	70	0	N
MED3	C	70	0	N
MED4	C	70	0	N
MED5	C	70	0	N
MED6	C	70	0	N
MED7	C	70	0	N
MED8	C	10	0	N
CHRONICDX1	C	70	0	N
CHRONICDX2	C	70	0	N
CHRONICDX3	C	70	0	N
CHRONICDX4	C	70	0	N
CHRONICDX5	C	70	0	N
CHRONICDX6	C	70	0	N
CHRONICDX7	C	70	0	N
CHRONICDX8	C	10	0	N
COMMENT	M	10	0	N
STUDYREP	M	10	0	N

Appendix V

Copies of Some E-mail Messages from Haley Enterprises

A Series of 16 E-mail messages was received from Haley Enterprises over a period of 7 months. In the end, they agreed to changed their source code for case-base text matching, and supplied us with an updated copy. The included E-mail messages are just some examples of the communication between UNB and Haley.

```
From v7h4@unb.ca Wed Jun 25 13:23:26 1997
Return-Path: info@haley.com
Received: from unb.ca (hermes.csd.unb.ca [131.202.3.20])
        by romulus.sun.csd.unb.ca (8.8.5/8.7.3/970527-09:15) with ESMTP id NAA12561
        for <v7h4@pop.unb.ca>; Wed, 25 Jun 1997 13:23:26 -0300 (ADT)
Received: (from root@localhost)
        by unb.ca (8.8.5/970625-08:00) id NAA21788
        for v7h4@pop.unb.ca; Wed, 25 Jun 1997 13:23:24 -0300 (ADT)
Received: from gate.usaor.net (root@gate.usaor.net [204.157.146.1])
        by unb.ca (8.8.5/970625-08:00) with ESMTP id NAA21720
        for <v7h4@unb.ca>; Wed, 25 Jun 1997 13:23:15 -0300 (ADT)
Received: from nottingham (dyn10cs10.usaor.net [205.199.35.42])
        by gate.usaor.net (8.8.5/8.8.5) with ESMTP id MAA25903;
        Wed, 25 Jun 1997 12:22:55 -0400 (EDT)
Message-ID: <33B14444.8482B417@haley.com>
Date: Wed, 25 Jun 1997 12:16:04 -0400
X-PH: V4.2.1@hermes
From: "W. Mark Boggs" <info@haley.com>
Organization: The Haley Enterprise
X-Mailer: Mozilla 4.0b4 [en] (WinNT; I)
MIME-Version: 1.0
To: v7h4@unb.ca
Subject: Re:
X-Priority: 3 (Normal)
References: <199706241900.QAA06502@jupiter.sun.csd.unb.ca>
X-Lines: 192
Content-Type: text/plain; charset="iso-8859-1"
Content-Length: 6056
Status: RO
X-Status:
```

Lijuan,

The code 70 is a codebase error that indicates that it could not open the file. We require a dBase IV file whose complete path is specified in the call to set_database_name. If you don't specify a full path, then it looks in the current working directory whatever it is.

The difference between CBR_SYMBOL_FIELD_TYPE and CBR_TEXT_FIELD_TYPE is how they are processed in the index. Symbols can be used in both decision tree and nearest neighbor indexes. In distance computations they either match, contributing a distance of 1.0 or don't match contributing a distance of 0.0. Text fields can be used in nearest neighbor indices and are processed in a variety of ways such as stemming and nmgramming. Symbols are only useful if the number of symbols in the domain is small. Either a memo or a character field can be used for either type, although it doesn't really make sense to use a MEMO field for a symbol.

I see that you have been looking at the phonebk.c example for guidance. I suggest that you get that example working under Visual Basic.

-Klaus

Klaus P. Gross, Ph.D.
The Haley Enterprise, Inc.
http://www.haley.com
Info@Haley.COM
(800) 233-2622, (412) 741-6420 voice, -6457 fax

```
From klaus@haley.com Mon Sep 15 22:26:51 1997
Return-Path: klaus@haley.com
Received: from unb.ca (hermes [131.202.3.20])
        by sisyphus.sun.csd.unb.ca (8.8.5/8.7.3/970630-13:55) with ESMTP id WAA01399
        for <v7h4@pop.unb.ca>; Mon, 15 Sep 1997 22:26:51 -0300 (ADT)
Received: (from root@localhost)
        by unb.ca (8.8.5/970625-13:40) id WAA19570
        for v7h4@pop.unb.ca; Mon, 15 Sep 1997 22:26:51 -0300 (ADT)
Received: from ivory.lm.com (ivory.lm.com [204.171.44.50])
        by unb.ca (8.8.5/970625-13:40) with ESMTP id WAA19564
        for <v7h4@unb.ca>; Mon, 15 Sep 1997 22:26:48 -0300 (ADT)
Received: from zeus (haley.slip.lm.com [204.171.41.200]) by ivory.lm.com (8.8.5/8.6.12)
with ESMTP id VAA18814 for <v7h4@unb.ca>; Mon, 15 Sep 1997 21:26:46 -0400 (EDT)
Message-ID: <341DDFB7.E3F00F20@haley.com>
Date: Mon, 15 Sep 1997 21:24:08 -0400
X-PH: V4.2.1@hermes
From: "Klaus P. Gross, Ph.D." <klaus@haley.com>
Reply-To: klaus@haley.com
Organization: The Haley Enterprise
X-Mailer: Mozilla 4.01 [en] (WinNT; I)
MIME-Version: 1.0
To: WANG <v7h4@unb.ca>
Subject: Re: About TER
X-Priority: 3 (Normal)
References: <Pine.SUN.3.96.970903080510.8646A-100000@jupiter>
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Status: RO
X-Status:

WANG wrote:

> Dear Klaus and Mark:
>
> Thank you for your reply.
>
> I tried to arrive at the result which I got from The Easy Reasoner
> based on the formula you gave me, but I still could not. The calculated
> similarity (0.2) between record 1 and query is far away from the result
> (0.7)the machine gave.

--> I apologize for the confusion, but the formula I gave you was for
similarity. To get the textual distance
between two records for a field, compute the similarity and then the distance =
max(1.0 - similarity,0.0)
Use this distance, which is guaranteed to be between 0 and 1.0, when combining
with other
fields.

> I checked the formula in detail and have several things confused:
>
> (1). nk for term "time" is 2. This means that only the number appearing
> in records are counted. However, in the query, there is also a "time". Why
> do we just ignore that?

--> Since the time term does not appear in the case base, it can never increase
the similarity.
See  "Developments in  automatic text retrieval",by Gerald Salton in Science,Vol
253,  (30 Aug,1991) pgs 974-980.
We used this reference to determine how to weight terms in a piece of text.

> (2). I see the formula is the same as the one in the manual
> except for the use of "EPSILON" (it is useful, indeed). I don't see any
> reason to use "LOG_OF_2"
> since it will cancel out when we calculate SUM and then W11, W12, W13
> (see following original message).
>
> (3). Is the "log" (log(x+EPSILON)...) to the base 10 or to the base e?
> Either way does not affect the calculated result much, but I would still
> like to know.

-->In C, log is base 10 and ln is base e

> (4). The index variable k (from 1 to T). The manual says T is the
> number of terms in the query. What you told me indicates that it is the
```

> number of terms in all records (not including the query). Confused really.
> Sorry about that.

The manual is incorrect, T is the nuber of unique terms across all recordsas my example indicated.

> (5). There is only 1 'Sum' in the similarity formula S(i,q) you gave me,
> but in the manaual, there are 2 'Sum's (see manual The easy Reasoner page
> 30-7). Either way, the calculated results
> don't match the result computed by the computer, but I would like to know
> the right formula.

--> The manual is incorrect, there should only be one sum. The purpose is tonormalize the similarity to fall bewteen 0.0 and 1.0.

> (6). Another question is for a field with date type or logical type in
> a database. What kind of index type do I need to specify for buliding an
> index using a date field type or a logical field type.

--> _cbr.h defines the constants CBR_DATE_FIELD_TYPE and CBR_BOOLEAN_FIELD_TYPE

> Could you please show me the source code use to calculate the similarity
> of two records with text fields? It will be very much appreciated.
> Looking at the source code appears to be the only way that I can be
> certain of what formulae are used when computing the distance.
>

--> I cannot do that, but I hope the above answers clarify the process. If not, please send mail again andI'll expand the example I gave you with actual numbers.

Please pardon the delay in responding,

--
Klaus P. Gross, Ph.D.
The Haley Enterprise, Inc.
http://www.haley.com
Info@Haley.COM
(800) 233-2622, (412) 741-6420 voice, -6457 fax

```
From klaus@haley.com Wed Sep 17 19:04:10 1997
Return-Path: klaus@haley.com
Received: from unb.ca (hermes [131.202.3.20])
        by sisyphus.sun.csd.unb.ca (8.8.5/8.7.3/970630-13:55) with ESMTP id TAA00561
        for <v7h4@pop.unb.ca>; Wed, 17 Sep 1997 19:04:08 -0300 (ADT)
Received: (from root@localhost)
        by unb.ca (8.8.5/970625-13:40) id TAA03267
        for v7h4@pop.unb.ca; Wed, 17 Sep 1997 19:04:06 -0300 (ADT)
Received: from ivory.lm.com (ivory.lm.com [204.171.44.50])
        by unb.ca (8.8.5/970625-13:40) with ESMTP id TAA02934
        for <v7h4@unb.ca>; Wed, 17 Sep 1997 19:03:02 -0300 (ADT)
Received: from zeus (haley.slip.lm.com [204.171.41.200]) by ivory.lm.com (8.8.5/8.6.12)
with ESMTP id SAA20131 for <v7h4@unb.ca>; Wed, 17 Sep 1997 18:01:16 -0400 (EDT)
Message-ID: <34205276.B6FA8053@haley.com>
Date: Wed, 17 Sep 1997 17:58:14 -0400
X-PH: V4.2.1@hermes
From: "Klaus P. Gross, Ph.D." <klaus@haley.com>
Reply-To: klaus@haley.com
Organization: The Haley Enterprise
X-Mailer: Mozilla 4.01 [en] (WinNT; I)
MIME-Version: 1.0
To: v7h4@unb.ca
Subject: TER example
X-Priority: 3 (Normal)
Content-Type: multipart/mixed; boundary="------------E6D744F5E75CA950039C4E2D"
Content-Length: 136244
Status: RO
X-Status:

This is a multi-part message in MIME format.
--------------E6D744F5E75CA950039C4E2D
Content-Type: text/plain; charset=us-ascii
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Content-Transfer-Encoding: 7bit

Lijuan,

S(i,q) = Sum(k=1,A wik*wqk)
D(i,q) = 1.0 - S(i,q).

For record 1, we have:

w11 = LOG_BASE_2(2/4)/3 = 0.333332
w12 = LOG_BASE_2(4/4)/3 = 0
w13 = LOG_BASE_2(1/4)/3 = 0.666665
sum = 0.555552
sqrt(sum) = 0.74535
w11 = .44721
w12 = 0
w13 = .89443

For the query, we have:
wq1 = LOG_BASE_2(2/4)/3 = 0.333332
wq2 = LOG_BASE_2(4/4)/3 = 0
wq6 = LOG_BASE_2(1/4)/3 = 0.666665
sum = 0.555552
w11 = .44721
w12 = 0
w16 = .89443

D(1,q) = 1.0 - S(1,q) = 0.8

I came up with 0.8 as well. That led me to look at the code more closely and as
a result I found a bug in the code.
I have attached a new version of cbr3ws.dll that fixes the problem.

Thank you for your persistance,

--
Klaus P. Gross, Ph.D.
The Haley Enterprise, Inc.
http://www.haley.com
Info@Haley.COM
(800) 233-2622, (412) 741-6420 voice, -6457 fax
```

VITA

Candidate's full name : Lijuan Wang

Place of birth : Shenyang, P. R. of China

Current address : 780 Montgomery St. Apt. 502
Fredericton, NB
E3B 2Y1, Canada

Permanent address : N/A

Universities attended : Northeastern University
Shenyang, P. R. of China
BSc(Math), 1983.

University of New Brunswick
Fredericton, Canada
MSc(Stat), 1995