AUTOMATIC ENCODING OF NATURAL LANGUAGE

MEDICAL PROBLEMS

A THESIS

Presented to

The Faculty of the Division of Graduate Studies and Research

by

Martha S. Hansard

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Information and Computer Science

Georgia Institute of Technology

November 1978
AUTOMATIC ENCODING OF NATURAL LANGUAGE
MEDICAL PROBLEMS

Approved:

Shannon D. Brunjes, Chairman

William E. Underwood

Fred R. Sias, Jr.

James Gough, Jr.

Albert N. Badre

Date approved by Chairman 10/13/1978
ACKNOWLEDGMENTS

I wish to gratefully acknowledge and thank the members of my thesis advisory committee for their exceptional support in the preparation of this dissertation. Dr. Shannon D. Brunjes, the chairman of my committee, provided many thoughtful criticisms, valuable suggestions, ideas, and resources essential for this research. He has given me loyal support and encouragement as well as the benefit of his wisdom and expertise. I especially want to thank Bill Underwood for his constructive criticism; he devoted much time and effort when he had many other important obligations. Fred R. Sias, Jr. also provided valuable insight with his careful critique. Sincere thanks are due to the other members of my committee, James Gough, Jr. and Albert Badre for their efforts in reviewing this thesis. Their comments and criticisms were very helpful.

I wish to express my gratitude to my family for their unselfish patience and encouragement. Dabney, my husband, encouraged me to seek my doctorate and has given me the support necessary to complete the degree. Dan and Ruby Snyder, my parents, have given me the strength to begin a task and to finish it.

I am indebted to my friends. The many discussions with Jack Corley, a colleague, were both stimulating and morally supporting. Marni Molak carefully edited this thesis and worked extra hours to complete drafts on schedule. Many other friends offered encouraging words at times when they were needed.

Thanks are also due to the staff of the MARIS project at the
Georgia Institute of Technology for supplying the test set problem-definitions. Their work is supported under NIH grant LM-02211 to the School of Information and Computer Science of the Georgia Institute of Technology and the Grady Hospital of Atlanta.

Finally, I would like to acknowledge the financial support of the National Library of Medicine through a NLM traineeship during the period of September 1975 through October 1978. This traineeship was a part of Public Health Service Training Grant, NLM-07014, to the graduate program in Biomedical Information Science in the School of Information and Computer Science.
TABLE OF CONTENTS

ACKNOWLEDGMENTS ...................................................... ii
LIST OF ILLUSTRATIONS .................................................. vi
SUMMARY ................................................................. ix

Chapter

I. INTRODUCTION ......................................................... 1

II. PURPOSE .............................................................. 3

III. CODING IN MEDICINE ............................................... 6

  Why Code Medical Natural Language?
  What is a "Good" Medical Code?
  What Medical Codes are Currently Being Used?
  Why are Current Coding Schemes Inadequate?
  Why Use the Anamnestic Coding Scheme?

IV. ENCODING IN MEDICINE .............................................. 29

  Why Automate Encoding?
  What Automatic Encoding Has Been Done?
  Natural Language Processing's Role in
  Automatic Medical Encoding

V. RESOURCES .......................................................... 54

  Design Set Problem-Definitions
  Past Work Using the Design Set
  Problem-Definitions
  Test Set Problem-Definitions
  Equipment

VI. METHODOLOGY ....................................................... 63

  Prepare Design Set
  Create New Dimensions
  Design and Implement Automatic Encoder
  Evaluate the Automatic Encoder's Ability to Produce
  Codes Agreeing with a Manual Encoder
  Evaluate the General Applicability of the Automatic Encoder
VII. DESCRIPTION OF THE ANAMNESTIC CODING SCHEME .... 75

- Meeting Criteria for Dimensions
- Groups of Dimensions
- Description of Dimensions
- Relations Among Dimensions

VIII. DESCRIPTION OF AUTOMATIC ENCODER ............ 106

- General Control Structure
- Description of Word Dictionary
- Description of Rule File

IX. RESULTS ............................................. 130

- Design Set Results
- Estimating the Value of the Word Dictionary Tests
- Estimating the Value of the Rule Section
- Results from a Test Set

X. CONCLUSIONS ........................................... 155

- Anamnestic Coding Scheme
- Mechanisms for Automatic Encoding

XI. RECOMMENDATIONS ................................. 166

- Improvements
- Implications for Future Work

APPENDIX .................................................. 169

REFERENCES ............................................. 172

VITA ....................................................... 180
LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-1</td>
<td>Summary of Automatic Encoding in Medicine</td>
<td>35</td>
</tr>
<tr>
<td>6-1</td>
<td>List of Standard Comments in CHCP Problem-definitions</td>
<td>66</td>
</tr>
<tr>
<td>7-1</td>
<td>List of Anamnestic Coding Scheme Dimensions with a Brief Description of Their Meanings in Problem-definitions</td>
<td>81</td>
</tr>
<tr>
<td>7-2</td>
<td>Sample Problem-definitions from the Design Set With Their Interpreted Anamnestic Coding Scheme Representation</td>
<td>96</td>
</tr>
<tr>
<td>7-3</td>
<td>Diagram of Logical Relations Among Some Dimensions</td>
<td>99</td>
</tr>
<tr>
<td>7-4</td>
<td>Diagram of Logical Relations Among Dimensions Containing Time Concepts</td>
<td>102</td>
</tr>
<tr>
<td>7-5</td>
<td>Diagram of Proposed Logical Relations Among Dimensions For a Coded Clinical Event of the Type Laboratory Test</td>
<td>104</td>
</tr>
<tr>
<td>8-1</td>
<td>Flow of Control in the Automatic Encoder</td>
<td>107</td>
</tr>
<tr>
<td>8-2</td>
<td>Options Available in Manual Mode</td>
<td>108</td>
</tr>
<tr>
<td>8-3</td>
<td>Flow of Control in the Preprocessor</td>
<td>110</td>
</tr>
<tr>
<td>8-4</td>
<td>Flow of Control Used to Automatically Assign Codes</td>
<td>113</td>
</tr>
<tr>
<td>8-5</td>
<td>Types of Numbers Recognized by the Automatic Encoder and Their Meanings</td>
<td>118</td>
</tr>
<tr>
<td>8-6</td>
<td>Sample Word Dictionary Entries</td>
<td>120</td>
</tr>
<tr>
<td>8-7</td>
<td>Categorization of Rules</td>
<td>128</td>
</tr>
<tr>
<td>9-1a</td>
<td>Summary of Encoding Results for Unique Problem-definitions in the Design Set</td>
<td>131</td>
</tr>
<tr>
<td>9-1b</td>
<td>Summary of Encoding Results for All Problem-definitions in the Design Set</td>
<td>132</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>9-2</td>
<td>Reasons for Considering Some Design Set Problem-definitions Manually Uncodeable</td>
<td>133</td>
</tr>
<tr>
<td>9-3</td>
<td>Design Set Problem-definitions Which Had Automatically Assigned Codes in Disagreement With Manually Assigned Codes</td>
<td>139</td>
</tr>
<tr>
<td>9-4</td>
<td>Design Set Problem-definitions Which Were Manually Codeable but Automatically Uncodeable</td>
<td>141</td>
</tr>
<tr>
<td>9-5a</td>
<td>Summary of Automatically Encoding the Manually Codeable Unique Design Set Problem-definitions Using Only the Most Frequent Word Sense Meanings in the Word Dictionary</td>
<td>143</td>
</tr>
<tr>
<td>9-5b</td>
<td>Summary of Automatically Encoding the Manually Codeable Design Set Problem-definitions Using Only the Most Frequent Word Sense Meanings in the Word Dictionary</td>
<td>144</td>
</tr>
<tr>
<td>9-6a</td>
<td>Summary of Automatically Encoding the Unique Design Set Problem-definitions With No Rule Directives</td>
<td>145</td>
</tr>
<tr>
<td>9-6b</td>
<td>Summary of Automatically Encoding the Design Set With No Rule Directives</td>
<td>146</td>
</tr>
<tr>
<td>9-7a</td>
<td>Summary of Automatically Encoding the Manually Codeable Unique Problem-definitions in the Design Set Using Only the Most Frequent Word Sense Meanings in the Word Dictionary and No Rule Directives</td>
<td>147</td>
</tr>
<tr>
<td>9-7b</td>
<td>Summary of Automatically Encoding the Manually Codeable Problem-definitions in the Design Set Using Only the Most Frequent Word Sense Meanings in the Word Dictionary and No Rule Directives</td>
<td>147</td>
</tr>
<tr>
<td>9-8</td>
<td>List of Test Set Problem-definitions Considered Manually Uncodeable</td>
<td>149</td>
</tr>
<tr>
<td>9-9a</td>
<td>Summary of Encoding Results for Unique Problem-definitions in the Test Set</td>
<td>150</td>
</tr>
<tr>
<td>9-9b</td>
<td>Summary of Encoding Results for Problem-definitions in the Test Set</td>
<td>151</td>
</tr>
<tr>
<td>9-10</td>
<td>List of Test Set Problem-definitions Which Were Manually Codeable but Failed to Automatically Encode Correctly</td>
<td>153</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>10-la</td>
<td>Number of Words in Design Set by Word Frequency</td>
<td>160</td>
</tr>
<tr>
<td>10-lb</td>
<td>Average Number of Word Sense Meanings in the Word Dictionary by Frequency of Words in the Design Set</td>
<td>161</td>
</tr>
<tr>
<td>11-la</td>
<td>Portions of the Current System Dimension Code With a Hierarchical Depth of Four</td>
<td>167</td>
</tr>
<tr>
<td>11-lb</td>
<td>Portions of an Improved System Dimension Code With a Hierarchical Depth of Six</td>
<td>167</td>
</tr>
</tbody>
</table>
SUMMARY

Traditional methods for recording medical data are inadequate for meeting current information retrieval demands. The Weed Problem-Oriented Medical Record (POMR) provides an organizational structure which facilitates computerization; however, input inconveniences and inadequate storage representations are major obstacles preventing computerization of medical records. Real-time, interactive, structured entry is possible for some medical data; but one portion of the POMR, the problem-definition, requires the expressive freedom available in Natural Language. A problem-definition is the physician's recorded statement of the patient's problem.

This investigation extended the Anamnestic Coding Scheme, which originated to code signs and symptoms and has the potential for being a machine representation for all information in the medical record, to code the semantic content of problem-definitions. A real-time, automatic encoder which maps problem-definitions stated in Natural Language into this semantic representation has been designed and implemented on a medium size minicomputer.

The current automatic encoder is a prototype computer program which allows the user to input a Natural Language statement of the patient's problem. If the encoder is able to assign a set of codes, it displays the codes with their interpretation. If the encoder is unable to assign a set of codes, it informs the user and indicates a reason for the difficulty. A large design set of problem-definitions guided the design of the automatic encoder and provided a useful corpus for
selecting concepts to be included in the coding scheme.

Current investigative results indicate automatic encoding of problem-definitions into the Anamnestic Coding Scheme is not only possible but beneficial for quality control. The automatic encoder achieved 99.9% agreement with manually assigned codes in the design set. It failed to code less than 0.1% of the manually codeable problem-definitions in the design set, and it correctly detected difficulties in 82.4% of the design set problem-definitions which were manually uncodeable.

With a test set of problem-definitions from a completely different patient population, the automatic encoder, with no modification, had 99.2% agreement with manually assigned codes, failed to code 20.1% of the problem-definitions which were manually codeable, and correctly detected difficulties with all problem-definitions considered manually uncodeable. The primary reason for failure to automatically encode problem-definitions in the test set was vocabulary differences between the design set and test set.
CHAPTER I

INTRODUCTION

Both the number of patients entering the health care process and
the amount of information kept on each patient increase every year. De­
mands for medical data retrieval exceed the abilities of teams of clerks
using traditional medical records (Priest, et al., 1978). There is a
pressing need to reorganize the medical record and use computer tech­
niques to process medical information requests (Gordan, 1970).

Natural Language text portions of medical records are a major
obstacle preventing full utilization of automatic techniques for mani­
pulating data (Bates, 1963; Kiely and Juergens, 1970; Schmidt, 1974;
Friedman and Gustafson, 1977). Health care providers will not exchange
the expressive freedom in Natural Language for restraints required by
most computer languages (Zielstorff, et al., 1977). Advances in Natural
Language processing indicate that translation from a restricted domain
of Natural Language to a machine representation is possible (Winston,
1977). However, studies must be conducted to discover sublanguage
nuances and define a machine representation.

The research described here is a study of an area of Natural
Language text within the Problem-Oriented Medical Record (POMR) called
the problem-definition (Weed, 1973). A problem-definition is a physi­
cian's statement of the patient's problem at the level at which the
physician understands it. According to Weed (1972) a problem is "anything
that requires management or diagnostic work-up; this includes social and
demographic problems." The POMR provides a logical organization of information within the traditional medical record since every data item within the POMR is linked to a medical problem, e.g., initial plans are made to treat or diagnose a problem and progress notes describe the state of a specific problem. A major advantage of the POMR is that all medical personnel can quickly grasp what has been done and is being done to manage the patient's problem.

Two general descriptions of problem-definitions indicate directions for Natural Language processing of problem-definitions. First, the content of a problem-definition varies greatly, e.g., it is more than a diagnosis. A problem-definition may be a diagnosis (diabetes), a symptom (dizziness), a physiological finding (heart failure), an abnormal lab test (hypersedimentation), or a social or economic problem (marital problem). The variety of content necessitates an expressive semantic representation for storage of all information on a computer. Secondly, the syntactic form of problem-definitions is relatively simple, i.e., problem-definitions are generally noun phrases (Pratt, 1973 and Mishelevich, 1972). There is no need for sophisticated parsing to recognize parts-of-speech such as verbs and objects. These syntactic and semantic descriptions of problem-definitions provide general directions for developing an expressive semantic representation (coding scheme) and developing a method of translating (encoding) from Natural Language to code.
CHAPTER II

PURPOSE

The objectives of this research are threefold:

(1) To extend the Anamnestic Matrix Coding Scheme (ACS) outlined by Brunjes (1971) to reflect the content of problem-definitions.

(2) To construct mechanisms for automatically encoding Natural Language problem-definitions into ACS.

(3) To project the feasibility of real-time encoding of Natural Language problem-definitions within a clinical environment.

ACS has the potential to encode all information in the medical record; no other medical coding scheme has such a potential. This research is intended to extend ACS to code all information in one small, important, portion of the POMR. Success here will contribute to automating medical records.

Achieving the second objective will overcome much of the medical community's resistance to automating the medical record. Many sections of the medical record can be structured for "menu selection" input. However, problem-definitions are too varied and numerous to input in this manner.

Achieving the third objective will provide the groundwork for
further development of an automated medical record system using ACS as a machine representation. It will also supply information necessary to implement an automated medical record system in a clinical environment. A successful automated medical record system which routinely collects and retrieves medical data will greatly benefit medical research.

Many resources are currently allocated for collecting medical data to explore special problems. After many years of data collection, researchers often develop hypotheses, not conclusions. The hypotheses need testing but the data which prompted formulation of the hypotheses cannot also be used to test them. Additional resources are then needed for controlled experiments to test hypotheses without bias. Thus the time from initial data collection to the use of knowledge gained in clinical medicine is intolerably long. If automated medical record systems are commonly used, at least exploratory data is available and many medical resources need not be wasted in unfruitful directions.

The following chapters discuss the background for this research, the methods used to achieve these objectives, and the results obtained. Depending upon his/her background, the reader may wish to select only portions of the following chapters for careful reading. The remainder of this chapter outlines the theme in each chapter.

Chapters III and IV discuss the background for the current research. Chapter III discusses medical coding schemes. Only the section "Why use the Anamnestic Coding Scheme" is necessary for later chapters. Chapter IV gives a history of automatic encoding in medicine and discusses the influence Natural Language processing had on this investigation.

The remaining chapters must be read, or at least skimmed, in order.
Chapter V describes two data sets used in this research. Chapter VI outlines the methodology. Chapter VII is an in-depth discussion of ACS. Chapter VIII is a brief description of the automatic encoder designed and implemented in this investigation. The reader may want to skim the formal discussion of the word dictionary and rule file and only read the last few paragraphs of these sections which give examples and describe the use of both the word dictionary and the rule file.

Chapter IX primarily describes the figures in that chapter. If the figures are self-explanatory, most of the text may be skimmed. Chapter X describes the conclusions made from the current work and Chapter XI provides some recommendations for future research in this area.
CHAPTER III
CODING IN MEDICINE

Why Code Medical Natural Language

The idea for coding Natural Language is not new in medicine. Reasons for coding are many:

1. To promote standard terminology;
2. To reduce storage costs;
3. To reduce retrieval costs;
4. To increase accuracy of retrievals;
5. To encourage completeness and consistency;
6. To facilitate multilingual and multidialectic communication;
7. To encourage use of automated data processing.

Elaboration of this list of reasons for coding follows in the next few paragraphs.

The medical profession is acutely aware of the need for standard terminology. In 1960, the American Medical Association (AMA) accepted the task of compiling a reference book of standard diagnostic terminology. Uniform terminology and definitions aid statistical analysis and information compilation in medical research and assist in allocation of health care resources (Gordan, 1973).

Coding promotes standard terminology because (1) it requires careful definition of terms prior to encoding, (2) the codes themselves may
be used as a representation for a standard terminology, and (3) all output generated from codes contains only standard terms which subtly influence usage.

Coding may reduce storage costs in both manual and computerized storage. Greenwood (1972) reported a maximum of 342 characters and a mean value of 19.4 characters per phrase from a sample of 40,020 morbidity phrases. The longest diagnostic coding scheme in common use has six fields of four characters each, i.e., a maximum of 24 characters per phrase. Three of the fields are rarely used concurrently to code a single diagnostic phrase. Although no information is available for average characters per coded phrase, the semantics of each field in this coding scheme dictate the average to be below 16 characters per phrase. In addition, the very nature of Natural Language communication with its inherent redundancies and circumlocutions provides strong evidence that any representation of content without these characteristics requires less storage. Gordan (1970) estimated medical record size, excluding charts and graphs, could be reduced 60-80% by simply using precise and specific terms. It is reasonable to conclude that coded diagnostic storage space is less than or equal to uncoded storage space.

Coding effects information retrieval in a number of ways. Assuming equal costs for storage of coded and non-coded data, one decision criterion is to code whenever

\[ C + (N \cdot Rc) < N \cdot Ru \]

where
\[ C = \text{cost of coding} \]
\[ Rc = \text{average cost of one coded information retrieval} \]
\[ Ru = \text{average cost of one uncoded information retrieval} \]
\[ N = \text{estimated number of retrievals} \]

This decision criterion does not consider several key issues. The first of these is that \( N \) may be a function of \( Rc \) or \( Ru \), i.e., the number of estimated retrievals may depend upon the cost of executing a retrieval. Secondly, the criterion assumes that retrieval involves a large number of records. With a small number of records an entirely different criterion would apply. Thirdly, the criterion assumes that the information sought is available in coded form for search of coded records and uncoded form for search of uncoded records. Finally, the criterion disregards retrieval benefit. A realistic decision model would include estimates for profit from retrieval. The following discussion is intended to explore the decision criterion for both an automated and manual retrieval.

For a manual information retrieval a clerk must (1) find each patient's record, (2) find pertinent information in the record, (3) decide whether the information is in the retrieval set, and (4) take appropriate action such as recording the information or pulling the record from the file. Variation in time for some of these manual steps depends upon whether coded or uncoded information is being retrieved.

In step one simply finding a patient's chart may be difficult or impossible. Anderson and Woodroffe (1969) reported difficulty finding 10% of all patient charts. Gallentine and Stuart (1973) found 13% to 16% of medical records unavailable at the time of patient visit at two U.S. Army hospitals. They noted that the percentages were inflated because
their figures included requests for medical records when patients had not made an appointment. Zielstorff (1977) reported 33% medical record unavailability in a one month study period in a nurse practitioner clinic. During the month only 13.7% of the visits were without appointment. For patients with appointments, the requests for medical records were made three days before a visit. Record availability does not depend directly upon whether information is coded or not coded in a manual retrieval. Therefore, one may assume that chart location times may be approximately equal for manual retrieval of both coded and uncoded information. However, the prevalence of misplaced charts does emphasize the need for improved medical record keeping.

In the second manual step, it is unlikely that all uncoded information will be in an orderly form ready for recognition. It is most likely that a trained clerk will be unable to read some information (Gordan, 1973) or have to scan completely through one or more pages in search of information. Gallentine, et al. (1973) reported four to five minutes to find diagnostic test results in a group of medical records at a large urban teaching hospital.

If coding is systematically done, no doubt the coded information will be easily found because the coding procedure will specify placing the code in a consistent location. Many manual coding procedures specify placing codes in a place other than the patient chart. For many years physicians have recorded identification numbers of patients with certain diseases. These records are essentially files inverted on diagnosis.

In the third manual step, decision time for coded information may be shorter than that for uncoded information. Theoretically, different
recognition response times for coded and uncoded information depend upon whether a human can scan a short sequence of characters faster than a long sequence. Without measuring recognition times, it is reasonable to conclude that manual decision time on coded data is no greater than manual decision time for uncoded data.

In the fourth manual step, the conclusion that taking appropriate action after finding coded information requires no more time than that for uncoded information is logical. However, the point must be made that appropriate action after finding uncoded information is, in many instances, simply to code that information.

The major cost difference between manual retrieval of coded versus uncoded information does not depend solely upon the time to do the tasks but rather upon the personnel required. Uncoded manual retrieval demands better trained clerks (higher paid) than coded manual retrieval. Therefore, under the assumptions of the previously stated decision criterion, the cost of manual retrieval for coded information is less than the cost of retrieval for uncoded information, i.e., \((R_{c\ manual} < R_{u\ manual})\).

For automated retrieval, the cost of retrieving information coded into a "good" coding scheme is clearly less than the cost of retrieving uncoded information. One factor which reduces costs of coded retrieval is a reduced number of character comparisons. Although information compaction is generally one result of coding, there are two other factors influencing retrieval costs. First, a "good" coding scheme includes some logical relations useful for retrieval. For example, body system may be the first character in a diagnostic code. For retrieval of all diagnoses involving one body system, only the first character of the code need be
compared. Second, synonymous information has only one code. There is no need to compare a search key with two synonymous phrases. With coded information, one comparison is enough. It is clear that computer storage of Natural Language text only postpones interpretive processing until retrieval time. One may conclude that the cost of automatically retrieving coded information is less than the cost for uncoded information (Re automated < Ru automated).

The conclusion that Re < Ru for both manual and automated retrieval, adds a constraint to the decision criterion. If Re = kRu where k<1, then the criterion becomes C + kNRu < NRu or C < Ru(N-KN). If 100 retrievals are estimated and the cost of a coded retrieval is half that of an uncoded retrieval, then an institution might well consider paying up to 50 times the cost of one uncoded retrieval for encoding the data. All the previous detailed explanation of how coding can reduce retrieval costs is capsulized in the fact that it is more economical to do a costly process (encoding) once than to do it repeatedly.

Coding may not only reduce retrieval cost, it may also increase retrieval accuracy. Manual retrieval is a tedious, repetitive task. It is quite possible that human retrievers make fewer errors when searching neatly presented codes rather than scattered, lengthy Natural Language text.

Computerized retrieval of uncoded data is generally performed by keyword. Precision and recall notoriously fall below 100% for keyword searches. Lamson, et al, (1965) stated, "It is the experience of many medical record libraries today that 75 percent success in a search for all cases desired by a physician user is a good yield." With a "good"
coding scheme, i.e., one that contains the information sought, accurately programmed automated retrieval guarantees improved recall and precision.

Another reason for coding is rather obvious but often overlooked. Coding encourages completeness and consistency. No matter how the coding is accomplished, the encoding process inherently involves a second judgment on the information which is coded. If information to be coded is so incomplete or inconsistent that the encoder recognizes a problem, then the encoder must either indicate the difficulty in the code or try to resolve the problem. Either solution may result in more consistent and/or complete retrieval than would otherwise have been possible for uncoded data.

Coding can make a definite contribution to communication among health care professionals who understand different languages. For example, the World Health Organization (WHO) compiles morbidity and mortality statistics from many nations with different native languages. Compilation requires translation into a common language. Codes could be the communication link. Moreover, a sophisticated encoder could allow different professionals who speak the same native language but have preferred linguistic idiosyncrasies to continue using their own special language while contributing to a common knowledge pool.

Finally, coding encourages the use of automated data processing. A few facts about the status of the health care industry and the role demanded of it in society emphasize the need for state-of-the-art information processing. First of all, society demands high quality health care at reasonable cost (Stagle, 1977). Medicine is in the midst of a data explosion (Norwood, et al., 1976). Shires reported terminology,
procedures, therapies, drugs, etc. increasing at an approximate rate of 10% per year in 1974. Jydstrup and Gross (1966) reported that handling costs for recorded information alone accounted for one-fourth of operating costs in the three hospitals they studied.

Rather than curtail the rising cost of medical record keeping by trimming the amount of data recorded, pressure is exerted to increase the amount of information retained on each patient. Threats of legal entanglement make health care providers order tests of doubtful diagnostic value and record details, not of medical interest, but of legal interest (O'Connor, 1976). Specialization and group practice also generate an abundance of data—much of it redundant (Jenken, 1973).

The enormous amount of medical data demands a rigorous organizational structure. As indicated previously, just locating a patient record may be difficult. Improved manual methods solve some data management problems; but utilization of a proven data management tool, the computer, promises further improvement. However, the medical community is today struggling with the problem of defining the computer's role in medicine.

Lincoln (1973) listed some startling reasons against using computers in medicine. Many items on Lincoln's "con" list were suppositions. However, he did cite instances where technology produced a social revolution which upset the roles of people and their vision of themselves and their surroundings. Lincoln asked, "If the medical record becomes an explicit statement of what is done, how do we handle mistakes?" He further surmised that the many failures of medical computing may have been "accidently on purpose" because one way to stop the introduction of a successful technology is to introduce an unsuccessful technology.
Whatever the reasons may be, medicine has not made full use of computer technology to manage the voluminous data within the medical record.

Some effort has been made to enter and store free text in the computer. This application of automation turns a complex machine into a filing cabinet. A computer may provide excellent indexing and improved communication switching for Natural Language text, but these functions are not optimum utilization of a costly tool. A "good" coding scheme, as defined in the next section, enables a computer to exhibit more than data manipulation. If the computerized data is stored in a logical form, then answers which are too costly to obtain may be found for many pertinent questions.

In order not to construe coding medical data as a panacea, it is necessary to present reasons why coding is not practiced universally in medicine. The following list states a few reasons for not coding medical data:

(1) Coding involves change.

(2) Coding all data in the medical record is difficult to justify unless it is used for research as well as primary patient care.

(3) Much coding benefit is delayed until everything in the medical record is coded.

(4) No "good" coding scheme exists.

Medicine is steeped in a tradition where changes occur very slowly. Codes offer a way to communicate that differs from traditional methods. Burgess Gordan, editor of the Journal of the American Medical Association,
stated in a 1972 editorial that American physicians were plainly against change solely for the sake of change. He further stated that changes can only occur in medicine when they are supported by substantial evidence for health care improvement. In other words, the general physician populace will only adopt new procedures if those procedures are debugged and have manifested worth for the physician or for the patient. The preceding discussion may show the worthiness of coding but successful implementation awaits a "good" coding scheme and encoding procedures.

Coding all data in the medical record is difficult to cost justify in pilot studies for debugging unless research as well as primary care utilize the data. In 1975, Collen, Van Brunt, and Davis reported that no successful total medical information (automated) system existed in the United States. Realistically, much quantifiable day-to-day patient care does not need sophisticated data manipulation, thus primary care communication can be conducted quite successfully in Natural Language. Unquantifiable benefits, such as improved health care resulting from structured quality control and numerous aggregate statistics, do not have direct impact on cost justification. An automated medical record (AMR) with much coded information requires a costly, total commitment. The only way to begin achieving sufficient expertise with AMR's is to include research benefits in cost justification.

It is difficult to cost justify coding small portions of the medical record in a working environment. The answers to many pressing questions require using information from many sections of the medical record. For example, finding common denominators among certain types of cancer patients necessitates using all information in each patient's record.
Coding one item in the medical record is a step forward but the benefits from this step cannot be evaluated fully until the remaining information in the medical record is coded.

Finally, no "good" coding scheme exists. The next section attempts to characterize a "good" coding scheme. The coding scheme itself is the most essential, yet hidden, element for cost justification.

**What is a "Good" Medical Code?**

A medical code is similar to any other code in that it communicates information from sender to receiver. The previous section established some reasons for using codes in medicine. For completely manual encoding and retrieval, a code is used primarily for conciseness and economy of communication among humans. When a code is stored on a machine, the primary reason for coding is to transfer information in a manner which allows the machine to execute a high level of manipulative power ("understanding").

Feinstein (1970) listed four coding scheme requirements to be used when encoding data for specific research projects. Clark (1974) reported nine necessary qualities for a good disease classification system. Shires (1974) gave eight criteria for a general medical coding scheme. Luff and Walker (1975) listed seven objectives for a practical coding scheme. Hopker, et al. (1977) presented one feature desirable in a medical coding scheme. The following composite list of paraphrases attempts to establish requirements for a medical coding scheme:

--Be acceptable to a majority of users (Shires, 1974; Clark, 1974)

--Be non-ambiguous, i.e., mutually exclusive
interpretations (Feinstein, 1970; Shires, 1974; Clark, 1974; Luff and Walker, 1975)

--Be an accurate and complete representation of the original meaning, i.e., convey desired shades of distinction (Feinstein, 1970; Shires, 1974; Clark, 1974)

--Allow for organization and categorization in a simple manner (Shires, 1974)

--Be flexible in adapting to change (Feinstein, 1970; Shires, 1974; Clark, 1974)

--Be reproducible (Shires, 1974)

--Contain provision for indefinite data or be accompanied by suitable criteria for coding such data (Feinstein, 1970; Clark, 1974)

--Conserve coding time and resources (Luff and Walker, 1975)

--Conserve computing resources (Clark, 1974; Luff and Walker, 1975)

--Be compatible with other existing coding systems (Clark, 1974; Luff and Walker, 1975)

--Have permanence in assigning codes (Luff and Walker, 1975)

--Provide for simplicity in coding, storage, and retrieval (Clark, 1974; Luff and Walker, 1975)

--Be used by all members of the health team (Clark, 1974)
What Medical Codes are Currently Being Used?

The forerunners of modern coding schemes date back to the seventeenth century (Huffman, 1972). Even at that early date there were distinctions between nomenclature and classification. A classification is a grouping of observations based on some criteria. A nomenclature is a list or catalogue of approved terms for describing and recording observations (Shires, 1974). The primary goal of nomenclatures is to establish standard terminology whereas classifications aid compilation of statistics.

Most attention has focused on classification and nomenclatures for mortality and morbidity. For many years, interest centered on collecting mortality statistics since sickness strongly implied death. As methods for treating illness improved, morbidity statistics received attention; however, the influence emphasizing mortality persists in modern classifications and nomenclatures.

Even though early efforts concentrated on mortality and morbidity, some researchers strove to classify or establish a nomenclature for such items as anatomy, surgical operations, or procedures. In several cases, operations were given codes in an appendix. Modern classifications and nomenclatures exhibit more integration of operations and procedures than did their predecessors.

The organization of classification and nomenclature schemes has evolved from simple lists to fairly complicated structures. One
nomenclature in 1869 contained 1053 disease terms (Huffman, 1972). A modern nomenclature contains 29,000 different disease names for 3,200 to 3,800 specific diseases. The necessity for rigorous organization is obvious. Along with alphabetical ordering, some schemes provide for order by system, etiology, or general disease class. The increasing need for mechanical retrieval has prompted the relation of specific code digits to a corresponding order structure.

Classifications and nomenclatures are still evolving; new terms and concepts are being added; structures are shifting; and periodic revisions are scheduled. Clearly, there is a need for breadth and flexibility in all coding schemes.

Shires (1974) listed the top seven classification and nomenclature schemes in North American hospitals with their estimated percentage of users:

54% ICDA-8 (International Classification of Diseases-Adapted for American use, version 8)
28% H-ICDA (Hospital version of ICDA)
15% SNDO (Standard Nomenclature of Diseases and Operations)
6% ICDA-7 (ICDA version 7)
4% CMIT (Current Medical Terminology)
3% SNOP (Systemized Nomenclature of Pathology)
1% CPT (Current Procedure Terminology)
4% Other personal or private code systems

The percentages reflect greater than 100% use because some institutions
utilize more than one coding scheme. Usage of SNDO has fallen due to the fact that the AMA discontinued revisions in 1961. The Standard Nomenclature of Medicine (SNOMED) does not appear on Shires' list since it is a recent addition to nomenclature schemes. In order to put coding in perspective, a brief review of the purposes and structure of each scheme is necessary.

ICDA-7, ICDA-8 and H-ICDA are all modifications of ICDA which originated with ICD. ICD began as a classification for causes of death. With the addition of morbidity classifications, ICD became suitable for hospital indexing of patient records. The American Hospital Association (AHA) modified ICD for detailed coding of hospital morbidity in the United States, and ICDA was the result. ICDA-7 and ICDA-8 are later revisions of ICDA. H-ICDA is a revision of ICDA which places greater emphasis on hospital use. All these ICD derivatives apply a basic three digit number to disease conditions. A fourth digit is used for specific conditions. A letter (E) before the four digit code enables coding external injury. A letter (Y) preceding a three digit code is employed for special conditions and examinations without sickness. A three character code is used for operations and nonsurgical procedures.

According to Huffman (1972), SNDO assumes "...every disease is the result of some cause acting in or upon an organ or tissue, and that every operation is a procedure performed on an organ or tissue." Consequently, SNDO codes have the form of a topographical code hyphenated with an etiological code. Topography codes to a minimum of three digits. Etiology codes to a minimum of three digits when it means a disease and two digits when a surgical procedure is indicated. A disease code number,
with topography, may have up to thirteen digits plus a behavior code letter. An operation code may consist of a maximum of ten digits. Both topographical and etiological codes are arranged hierarchically with the leftmost digits being general and the rightmost, specific.

CMIT and CPT are sponsored by the AMA. A CMIT coding manual contains disease names arranged alphabetically. Each disease entry contains subparagraphs of text detailing synonymous terms, etiology, symptoms, physical signs, complications, laboratory data, X-ray data, and pathology data. Only disease names are coded with a six digit number; the other information pertaining to the disease has no codes associated with it. The first two digits of a disease code represent the major topographic groupings of SNDO. The remaining four digits are randomly generated.

CPT is the procedure counterpart of CMIT and was developed along lines similar to CMIT.

The College of American Pathologists sponsors SNOP; therefore, it is arranged to suit detailed pathology descriptions. SNOP has four dimensions—topography, morphology, etiology, and function. Each field has a four digit code; however, X's and Y's expand the range available with digits. The codes provide a hierarchy with the leftmost digits general and the rightmost tending to be specific.

SNOMED is the newest coding scheme to meet with general acceptance. It uses the four dimensions of SNOP and adds two fields, procedure and therapy. Promoters of SNOMED intend it to be more applicable to general medical data than SNOP is.

The World Health Organization (WHO) issued a trial version of the International Classification of Diseases-Oncology (ICD-0) in 1975,
therefore it was not mentioned in Shires (1974). ICD-0 was derived from the American Cancer Society's Manual of Tumor Nomenclature and Coding (MOTNAC). Both ICD-0 and MOTNAC are a hybridization of the Neoplasm chapter in ICD and the Morphology category for neoplasms in SNOP. ICD-0 is intended to fulfill the needs for detailed cancer classification in tumor registries.

The class "other" on Shires' (1974) list contains numerous coding schemes, each with few users. For example, the Multinomat Foundation for Medical Care in Portland, Oregon, contracted the Dikewood Corporation to provide system analysis and data processing expertise for an Experimental Medical Care Review Organization (MEMCRO) grant (Luff and Walker, 1975). Their study showed existing codes inadequate for the subjective and objective findings which precede medical evaluation. Subsequently, they devised their own coding scheme based on the work of Brunjes (1971). Other groups, primarily those involved in research have originated their own coding schemes. The conclusion that existing, widely used coding schemes are not sufficient for all users is obvious. The next section details the inadequacies in existing coding schemes.

**Why are Current Coding Schemes Inadequate?**

Very few criteria for a coding scheme listed in a previous section apply to currently used medical coding schemes. Details of the many inadequacies of current coding schemes have been described elsewhere (Gordon, 1972; Lister and Cameron, 1974; Luff and Walker, 1975; Graepel, 1976). In this section, two major inadequacies of existing coding schemes will substantiate the claim that better coding schemes need to be developed.
Current classifications and nomenclatures are not acceptable or appropriate for a majority of users. The number of specially designed coding schemes or modifications of existing coding schemes is clear evidence that users find existing coding schemes inadequate (Morgan, 1971; Schmidt, 1974; Luff and Walker, 1975, Priest, et al., 1978). Classification schemes are by definition intended for a specific user group. Indeed, experience indicates that a universal disease classification may never be reasonable (Gordan, 1972; Clark, 1974). Nomenclatures are equally biased toward the interests of their sponsors. Traditional nomenclatures and classifications are simply inadequate for serving a general user community.

One additional characteristic of current coding schemes will suffice to close this discussion of inadequacies. That characteristic is the limited domain of coded information. McFarlene and Norton (1973) reported measuring coding scheme adequacy by the percentage of diagnoses which did not fit any code. They found 25% of the diagnoses made over a four year period in the hospital they studied did not fit any ICDA category. Westbury and Tarrant (1969) compared the capability of ICD-7 and the Royal College of General Practitioner's coding scheme (C.GP.C) to code problems occurring in 10,191 doctor-patient contacts in a general practice setting. They found ICD-7 did not cover 23% of the information included in patient problems and 45% of the problems could not be correctly coded into C.GP.C.

Current medical emphasis is on ambulatory care and preventive medicine (Fokkens, 1976). The POMR furnishes an adequate structure for recording these medical events. However, established classifications and
nomenclatures cannot reflect the information encountered in problem-
definitions which may be more than a diagnosis, procedure, or therapy.

Why Use the Anamnestic Coding Scheme?

The medical record may be viewed as a set of clinical events. Each clinical event is any single incident which has potential, direct impact upon the health of a patient. The Anamnestic Coding Schemes (ACS), as conceived by Brunjes (1971), is a multidimensional representation of concepts found in clinical events. Dimensions contain what might be called "semantic units" or "primitives" organized hierarchically. For example, the respiratory system includes lungs and bronchii within the System dimension. Therefore, the lungs and bronchii are said to be "lower" in the System dimension hierarchy than the respiratory system.

Every coded clinical event, when coded into ACS, is a unique point in n-space. The uniqueness property allows usage of an ACS representation for primary care because all descriptors of the coded clinical event are coded into dimensions which uniquely and completely specify each clinical event, i.e., all information in a problem definition stated in Natural Language is present in the ACS coded representation of that clinical event. Theoretically, a decoder, which mapped from an ACS representation into a Natural Language statement, would produce a close paraphrase of the statement in its original form.

No attempt has been made in this investigation to measure the completeness of information captured in ACS. There are two possible ways to obtain such a measure. One way involves obtaining subjective judgments from a panel of physicians as to whether an original problem-definition
and its decoded paraphrase convey the same information. Another way involves comparison of inferences which can be made from both the original Natural Language statement and an ACS representation of that statement. Both these experiments involve resources which are not available at this time. The first experiment necessitates constructing a decoder which is a large project in itself. The second experiment requires inference mechanisms which need information obtained in this current investigation. In other words, the results obtained here are necessary for further evaluation of ACS.

ACS is designed to be used with a fully automated medical record system with input made close to the source of information to reduce transcription errors. Many portions of the medical record may be input via "menu selection," i.e., the computer displays a set of options and the user merely selects what he/she wants to input from these. With this structured type of input, encoding into ACS is a simple mapping process. The current investigation provides a means to automate one medical record section, the problem-definition, where "menu selection" is not appropriate due to the large amount of information which would have to be displayed. Under this plan of input, an automated medical has guaranteed encoding consistency.

ACS is especially useful for retrieving patient information in many different forms. For example, all problem-definitions dealing with the respiratory system can be found by searching with a single key. In addition, n-space may easily be collapsed on one or more dimensions for aggregate statistics, which include summaries for one patient or many patients. After collapsing (reducing the number of dimensions), synonyms
map into identical points in the reduced space.

This investigation makes no attempt to evaluate retrievability of information coded into ACS for several reasons. First, a complete database system is necessary for such an evaluation. Secondly, some preliminary work at Yale indicated that ACS was very useful for retrieving medical information (Finseth, Dallal, Freeman, and Brunjes, 1977). Finally, experience indicates that medical researchers, when asked to project their information retrieval needs, tend to state only requirements with which they are familiar. Often they do not expand their information retrieval requests to take advantage of computerization, especially when they are unfamiliar with the computer's capability (Groner, Palley and Shapiro, 1975). Therefore, surveys of information needs fail to meet their objectives. The only way to measure retrievability is to implement a system, teach basic system uses to researchers, and measure how successful the system is in answering requests over a long period of time.

ACS avoids many of the disadvantages inherent in currently used coding schemes. It is not a classification or a nomenclature designed for a select set of users. It attempts to represent meaning in a manner useable for diverse purposes and it also does not restrict the type of coded information to one type of medical statement, e.g., diagnosis.

ACS is especially adept at coding vague information found in POMR problem-definitions. Indefinite information is acceptable in the POMR since the physician, following Weed's guidelines, records only his/her understanding of the problem at the time the problem-definition is made. In other words, the physician does not include any suppositions in the
problem-definition. Much of medicine is devoted to diagnosing; therefore, many problems are only vaguely understood at initial encounters. The hierarchical structure of ACS provides the ability to code and categorize both indefinite information and very specific information because indefinite concepts are at higher levels and definite concepts are at lower levels.

If all projections for ACS can be demonstrated to be feasible, then ACS will be an acceptable coding scheme for all users. The reasoning behind this statement is that no user need ever see the coded representation. An encoder and decoder will be interfaces between the user and the stored representation. The user can continue to converse in Natural Language. Of course, the consistency inherent in mechanized encoding and decoding will apply a subtle influence upon the user to be concise in constructing Natural Language statements; but this is an extra benefit of automation.

Thus far in this section, ACS has been shown to meet or have the potential for meeting all but two of the criteria for a "good" coding scheme listed in this chapter under, "What is a 'Good' Medical Code?" The remaining criteria are, "Be compatible with other existing coding systems" and "Have permanence in assigning codes." Theoretically, ACS could meet the first of these because ACS stores all information. A translation to another coding scheme would only mean collapsing some dimensions and mapping into the other coding scheme.

The second, and last, criterion to be discussed has questionable worth when all medical record information is stored on a computer and there is an algorithmic way to change codes. When Luff and Walker (1975) specified the need for permanence in code assignment, they were undoubtedly considering the difficulties when a revision of ICD or SNOP is released.
With a change in coding schemes, certain codes from consecutive years are incompatible and it is very expensive to manually recode all data from previous years. Changing codes in automated systems may also be expensive; but changes may be made quicker on automatic systems than in manual systems and changes made automatically are consistent.

The current investigation builds upon previous work which showed ACS to be a promising machine representation for clinical events. Lynch (1975) used ACS for coding signs and symptoms. Powsner (1978) indicated the appropriateness for two dimensions of ACS with problem-definitions. The current research develops additional ACS dimensions to code the information in problem-definitions and refines the structure of the two dimensions previously developed.

An important aspect of Lynch and Powsner's work was empirical development of dimension structure. They did not hypothesize a structure and then fit the data to the structure. Rather, they made an initial hypothesis based on observation, then iteratively modified their hypothesis to suit the data. The current development of ACS continues to be of an empirical nature by extracting the concepts in a large set of problem-definitions and organizing these concepts into a logical structure called ACS. For reasons cited above, many benefits of ACS are not evaluated; however, the contribution made in this research makes evaluation possible in the future.
CHAPTER IV

ENCODING IN MEDICINE

Why Automate Encoding?

There are five major reasons for automatically encoding medical text:

(1) To improve correctness and consistency in assigning codes;
(2) To reduce encoding costs;
(3) To facilitate error checking;
(4) To increase encoding adaptability;
(5) To facilitate health care provider communication with an AMR system.

The first reason may be restated as: When algorithms can be defined, machines perform repetitious tasks with more correctness than humans. Therefore, certain automatic encoders achieve fewer errors than manual encoders. Several studies comparing manual and automatic encoding have shown this to be the case with encoding diagnoses.

Howell (1971) compared manual and "fruit-machine" encoding into ICDA-8 on 25,177 hospital discharge diagnoses in 1968. (For a description of the "fruit-machine" see the next section.) Codes had been routinely assigned to all diagnoses by clerks as part of standard morbidity reporting. Howell found 3.5% errors in machine coding and 17.2%
errors in manual code assignment. In addition, the machine did not code 4.6% of the diagnoses for reasons ranging from unclear diagnosis to inadequacy of the machine dictionary. Howell reduced the workload of deciding code correctness by arbitrating only when manual and "fruit-machine" codes disagreed. This made the study feasible but it also ignored the possibility of both encoding modes arriving at the same wrong code.

Dinwoodie and Howell (1973) compared manual and automatic encoding of 1,512 diagnoses from sixteen general practitioners. A single clerk had manually encoded all 1,512 diagnoses into ICDA-8 and these codes were compared with those from the identical "fruit machine" encoder cited in Howell's 1971 article. Results were 0.2% machine errors, 12.3% manual errors, and the machine failed to code 4.2%. Dinwoodie and Howell concluded that the "fruit-machine" encoder also produced more correct codes on general practice diagnoses than did manual encoding.

Greenwood (1972) modified Howell's "fruit-machine" encoder. (For a description of the modification see the next section.) Greenwood compared manual and automatic encoding on two sets of hospital diagnoses. The first set consisted of 952 diagnoses with ICDA-8 codes assigned by clerks at various hospitals. The second set of 882 diagnoses was also obtained from several hospitals, but ICDA-8 codes were manually assigned at a central reporting facility. Results from the first set showed 2.4% automatic encoder errors, 20.1% manual errors, and the automatic encoder failed to code 13.7%. For the automatic encoder, 10.2% of the diagnoses were not encoded because words were not in the dictionary. For the second set of data, there were 1.9% automatic encoder errors, 8.16% manual errors, and the automatic encoder failed to
code 18.4%. Again, automatic encoder failures due to dictionary deficiencies were high, 14.7%. Greenwood concluded that, with a more complete dictionary, the automatic encoder would be more successful at assigning correct codes than were manual encoders.

Dinwoodie, Howell, and Greenwood all argued that manual encoding was a tedious task prone to human errors; furthermore, automatic encoder errors were consistent and easily correctable. Current manual encoding has difficulty being consistent because often rather arbitrary code assignments are made into existing coding schemes. For example, Howell (1971) discovered inconsistent duplication in the manual coding of patients with two circulatory diseases. At times these were coded as with hypertension and at other times without hypertension. The amount of coordination and memory required for a team of manual encoders to encode consistently is more than can be reasonably expected from humans performing a routine task. When a machine encodes, consistency is inherent.

The second reason for automatic encoding, to reduce encoding costs, is closely related to the first. Undoubtedly, under certain circumstances, automatic encoding is less costly than manual encoding. Cost estimates must consider equipment and software development; but correctness of code assignment is of equal, or greater, importance. Priorities must be established between accuracy and cost.

Myers and Hendrickson (1973) found many errors in manually coded diagnostic tests when they studied a subset of Professional Activity Study (PAS) abstracts from a large urban hospital in Pennsylvania. PAS manual encoding of diagnostic tests differs from diagnostic encoding into, for example, ICD because PAS uses a fairly short marksense form.
for tests. Thus PAS test encoders do not have to consult an entire volume to arrive at a code. Since PAS test encoding is much simpler than encoding a diagnosis, one would assume that there would be fewer errors. Myers and Hendrickson reported to the contrary. They found the major cause of errors was the encoders' lack of knowledge about hospital procedures. They concluded "...the nature of the work is such that it is unlikely that people with qualifications to approach theoretical accuracy would agree to serve as coders." Therefore, for a given level of accuracy, it may be less expensive to automate encoding rather than hire highly trained people.

The third reason for automatic encoding is again closely related to correctness and consistency. Automatic encoding facilitates error detection. There are at least two types of encoding errors. One type is random erroneous code assignment. A Public Health Service study (1974) of data quality in Vermont hospital reporting systems found that diagnoses reports from Vermont hospitals included five male hysterectomies and twelve females with operations on male genitalia. The report indicated that some errors could be detected by simply checking whether the code was a valid number in the coding scheme. It was estimated that less than 40% of all ICDA diagnoses could be checked by age/sex rules and approximately 15% of hospital procedure codes could be checked by diagnosis. Although such error checking does not discover all random errors, a well-designed automatic encoder has the distinct advantage of using all information at its disposal to routinely check code assignment. Of course manual encoding can also provide for error checking. The key factor is availability of additional information. For environments where
automatic encoding is practical, it is likely that an abundance of patient data will be available for use in routines for random error checking.

The second type of error results from insufficient information at encoding time. Manual encoders rely on their best judgment if data essential for correct encoding is missing. Automatic encoders can have the same decision problems; however, automatic encoding is more conducive to encoding at a time closer to data generation than is manual encoding. If someone close to the source of data enters a Natural Language statement that has unrecognized meaning, ambiguous meaning, or missing facts, the computer can automatically request additional information. Few physicians or nurses, who are the primary sources of medical data, have the time or inclination to manually encode the statements they insert into a medical record. It is possible that they might consent to input free-text or at least the clerk who transcribes data could use a computer terminal to enter free-text. Either way, the information is captured close to its source and the computer can prompt for additional information; whereas a manual encoder in a hospital's Medical Record section is limited to using only the information recorded at some time in the past.

The fourth reason for automatic encoding is adaptability. Computer programs can be changed and those changes communicated rapidly. Howell (1971) reported that many manual errors in his study were possibly due to the timing of his data collection. During 1968, when data was collected manual encoders were changing from ICDA-7 to ICDA-8. Some of the manual errors due to the unavailability of adequate ICDA-8 information. With an automatic encoder, changes in coding scheme would only require program or dictionary modification. With manual encoding, communication
of changes involves reprinting several volumes and training many people. In addition, all historical text can be quickly encoded again with automatic encoding. This is especially helpful if program changes generate bugs, but it also facilitates coding additional information not previously considered. No manual encoding has the prerogative of reexamining all previously assigned codes.

Finally, automatic encoding of medical text provides for the most sophisticated direct communication link between health care provider and computerized medical record. If the health professional could communicate with the computer in his Natural Language, then he would have all the computer's power at his immediate disposal without necessitating a third party communication. A successful automatic encoder of medical text would solve many input problems.

**What Automatic Encoding Has Been Done?**

Table 4-1 summarizes some automatic encoding efforts described in the literature. Coding or classification schemes in this table fall into three major categories: existing, hybrid, and created. Existing codes are ICD, ICDA, H-ICDA, SNOP, SNDO, and SNOMED. Hybrid codes are selected portions of existing codes which best fulfill needs and/or additional invented schemes. Created codes are invented for the occasion. They may be either conceptualized and/or derived from a working corpus.

Requiring special mention are systems which store complete Natural Language text and retrieve by keyword. Okubo, Russel, Dimsdale, and Lamson (1975) at UCIA reported coding words of entire text into integers to conserve space. They constructed a thesaurus of synonyms and hierarchical forms of predecessors and successors. Search were conducted
Table 4-1. Summary of Automatic Encoding in Medicine

<table>
<thead>
<tr>
<th>SYSTEM ID</th>
<th>DATE</th>
<th>PLACE</th>
<th>CODE</th>
<th>CORPUS</th>
<th>MEASURE</th>
<th>LANG/COMP</th>
<th>METHOD</th>
<th>DICT SIZE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. J. Smith, J. Melton</td>
<td>1963</td>
<td>Western Reserve University, Cleveland, Ohio</td>
<td>SHRO for topography; invented hierarchy of pathologic processes, modifier, Etiology, Injury, Operation, Non-specific site</td>
<td>36 sample autopsy diagnoses</td>
<td>Using corpus, 13 questions asked, all but 1 had 100% agreement with manual retrieval</td>
<td>GE 225</td>
<td>Words in sample were categorized &amp; coded; no mention of ambiguity; adjectival noun coded as noun; can specify logical relations in queries; queries require specialist</td>
<td>1,250</td>
<td>When comparing manual and computer retrieval on 56 records; manual was added to after computer output seen</td>
</tr>
<tr>
<td>2. Thatch</td>
<td>1964</td>
<td>U.S. Army</td>
<td>Army Surgeon General Code Book, SHRO for Etiology and Topography</td>
<td>Clinical record summary sheets</td>
<td>N/A</td>
<td>IBM</td>
<td>Each letter in word used to get a unique numeral; numeral replaces word and is locked up; lookup returns class (etiology, topography, circumstances of injury); word identified as major or minor; numerals encoded by class, added for each word and sum is code</td>
<td>4,553</td>
<td>Word sums got down to .44 redundancy</td>
</tr>
</tbody>
</table>
### Table 4-1. Summary of Automatic Encoding in Medicine (con't)

<table>
<thead>
<tr>
<th>SYSTEM ID</th>
<th>DATE</th>
<th>PLACE</th>
<th>CODE</th>
<th>CORPUS</th>
<th>MEASURE</th>
<th>LANG/CMP</th>
<th>METHOD</th>
<th>DICT SIZE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>J. B. G. Lamson, 1965</td>
<td>UCLA</td>
<td></td>
<td>Unique integer for each word</td>
<td>Surg Path</td>
<td>False positives—all areas; Dec 1970-1974 5.90% of all cases retrieved; For 1974 reduced to 3.49%</td>
<td>PL/1 Format translator as assembler</td>
<td>Entire text coded word for word to integer, retrieval by keyword match with logical operators, thesaurus containing synonyms, prepositional and possessive, and logical relations consulted for retrieval, specialist needed to set up queries.</td>
<td>March 1975</td>
<td>False positive causes Dec 1970-Dec 1974; Data base-25%, thesaurus links-45%, request form 54%, program 9%</td>
</tr>
<tr>
<td>B. Dinsdale, 1975</td>
<td></td>
<td></td>
<td></td>
<td>Bone Marrow, Autopsy, Nuclear Medicine, Neurology</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R. A. Orudo, 1975</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W. S. Rundell</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. ACDMS—Automated Code</td>
<td>1967</td>
<td>Roocewell</td>
<td>Rocewell</td>
<td>Surgical report narrative</td>
<td>1 request about position</td>
<td>---</td>
<td>Dictionary, lookup—part-of-speech, name number, subroutines to resolve ambiguities and handle idioms. Determine dependencies by Harris callation automation. Pick kernels from dependencies.</td>
<td>2700 words</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1969</td>
<td>Park</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1972</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Memorial Institute, Buffalo, New York</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. R. W. Howell</td>
<td>1968</td>
<td>---</td>
<td>ICD</td>
<td>Hospital discharge, mortality</td>
<td>25,000 discharge:</td>
<td>&quot;fruit-machine&quot; code</td>
<td>171 discharge diagnoses too poor to code; 358 fail due to abbrev; 140 had no agreed upon code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H. P. Dimond, 1971</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R. M. Loy</td>
<td>1973</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4-1. Summary of Automatic Encoding in Medicine (cont)

<table>
<thead>
<tr>
<th>SYSTEM ID</th>
<th>DATE</th>
<th>PLACE</th>
<th>CODE</th>
<th>CORPUS</th>
<th>MEASURE</th>
<th>LANG/CONF</th>
<th>METHOD</th>
<th>DICT SIZE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. KODIAC; R. N. Greenwood</td>
<td>1971</td>
<td>Medical Research Council, London</td>
<td>ICD</td>
<td>Morbidity and mortality phrases 952, 999, 842</td>
<td>86% to 93% for 3 diff sets of phrases</td>
<td>Fortran, some assembler</td>
<td>XDS 9100</td>
<td>6492 terms</td>
<td>Principle failure was unrecognizable terms due to abbreviations and misspellings</td>
</tr>
<tr>
<td>7. PARSE: R. L. Moog, P. Gagnon</td>
<td>1971</td>
<td>Univ of Ill</td>
<td>No code; just key of site diagnosis and modifier; pointer into hierarchical taxonomy</td>
<td>600 GYN, Version 1 67%, pathology diagnostic statements 615 statements, Version 2 91% diagnostic statements 493 statements</td>
<td>PL/1 IBM 360</td>
<td>62 delimiters, based on word morphology and syntax, fall into 13 categories used to form MOV attack stack then fits one of five major formats or variation identify site, diagnosis, and modifier</td>
<td>62 delimiters (13%)</td>
<td>Error types: 1) Delimiter list too short (35%) 2) Complicated sentence (17%) 3) Multiple sites (15%) 4) Misspelling of delimiters (13%)</td>
<td></td>
</tr>
<tr>
<td>8. MEANINGEX: D. E. Mihalovich</td>
<td>1972</td>
<td>Johns Hopkins</td>
<td>Semantic tree of modifiers, &quot;sparse,&quot; head term is noun</td>
<td>problem definitions</td>
<td>Assembler</td>
<td>CDC 3300</td>
<td>1) lexical, part-of-speech, RG standard terms 2) syntax prepositional phrases and normalization 3) Implications list directs building of sparse</td>
<td>NG</td>
<td>No actual use described; implication list beyond abilities at this time</td>
</tr>
<tr>
<td>SYSTEM ID</td>
<td>DATE PLACE</td>
<td>CODE</td>
<td>CORPUS</td>
<td>MEASURE</td>
<td>LANG/COMP</td>
<td>METHOD</td>
<td>DICT SIZE</td>
<td>COMMENT</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>--------</td>
<td>-------------------------</td>
<td>---------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>1973 MIN</td>
<td>SHOP</td>
<td>Surgical autopsy, and</td>
<td>NG</td>
<td>PL/1 IBM 360</td>
<td>If necessary, transform by morphosyntactic analysis from adj to noun and reorder words; lookup words or phrases to get TMEY</td>
<td>15,000</td>
<td>Simple match of SHOP statement encodes only 25%</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>1975 McGill Univ</td>
<td>N-ICDA for diagnosis, SHOP for</td>
<td>Surgical pathology</td>
<td>NG</td>
<td>PL/1 IBM 360</td>
<td>Preedited; must match phrase in dictionary</td>
<td>4--discard, Diagnosis and Anatomy, Topography, Neoplasia</td>
<td>Used to print reports, no retrieval use described</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>1975 New York 1976 Univ</td>
<td></td>
<td>Format of word classes derived by word use clustering</td>
<td>159 x-ray</td>
<td>N/A</td>
<td>1) Linguistic string analysis for sentence structure 2) Regularization by transformation 3) Mapping into format slots</td>
<td>N/A</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>1976 MIN</td>
<td>Greepel</td>
<td>User determined subset of</td>
<td>NG</td>
<td>PL/1 IBM 370</td>
<td>Where possible, all words transformed to nomitive singular, rest considered modifiers; noun lookup in dict; dict tells when modifiers need to be used</td>
<td>Estimate 2000-4000 entries</td>
<td>Purpose to utilize some principles of Pratt's system and to develop user-oriented reference dictionary based on SHOP principles</td>
<td></td>
</tr>
</tbody>
</table>
Table 4-1. Summary of Automatic Encoding in Medicine (cont)

<table>
<thead>
<tr>
<th>SYSTEM ID</th>
<th>DATE</th>
<th>PLACE</th>
<th>CODE</th>
<th>CORPUS</th>
<th>MEASURE</th>
<th>LANG/CMP</th>
<th>METHOD</th>
<th>DICT SIZE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>13. Panichell, Barnett</td>
<td>1976</td>
<td>Mass Genl Hoop</td>
<td>ICD-9 for diagnosis; SNOP for topography</td>
<td>Application</td>
<td>Exact match 68%; matched by disinfection 1%; matched after spelling, correction 3%</td>
<td>Hoop 11</td>
<td>Steps: 1) exact match, 4,000 terms 2) disinfection then match, 3) correct spelling (soundalike) then match, 4) rules</td>
<td>Rules: 1) Multiword phrases which go to one code, 2) words which must be adjacent or in a certain order, 3) multiword references converted to numbers; not much detail on rules</td>
<td></td>
</tr>
<tr>
<td>14. PAIRS-Pathology Information Retrieval System; L. A. Scudiero, R. L. Wong</td>
<td>1976</td>
<td>Univ of Ill</td>
<td>Devised Operative procedures and organs code</td>
<td>16,002 operative procedure sentences</td>
<td>Could not code procedure and/ or site in %; precision 92% recall 92%</td>
<td>PL/I IBM</td>
<td>Dict lookup flags; keywords, replaces synonyms, adds related words and corrects spellings; code assigned to each keyword</td>
<td>NG</td>
<td>Reasons for no code were ambiguity, missing info or inadequacy of code</td>
</tr>
<tr>
<td>15. B. Radin, C. Schade, C. Speck</td>
<td>1976</td>
<td>Baylor College of Medicine</td>
<td>Devised 200 categories for problems</td>
<td>20,200 problem statements</td>
<td>12% not coded from a sample of every 10th one; 7% correct; 9% incorrect</td>
<td>----</td>
<td>Compress by deleting NG vowels; enter table with “lead term”; table specifies substitution term comparisons for getting code; codes then compared for conflict and one with highest precision selected</td>
<td>Description in article inadequate for understanding</td>
<td></td>
</tr>
</tbody>
</table>
Table 4-1. Summary of Automatic Encoding in Medicine (con't)

<table>
<thead>
<tr>
<th>SYSTEM ID</th>
<th>DATE</th>
<th>PLACE</th>
<th>CODE</th>
<th>CORPUS</th>
<th>MEASURE</th>
<th>LANG/COMP</th>
<th>METHOD</th>
<th>DICT SIZE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>16. W. White, B. Barkman, L. Barriere-Bonneville, L. Cousineau</td>
<td>1977 Univ Hosp of Sherbrooke (French)</td>
<td>Trial version of SHOMED</td>
<td>French ONGYN</td>
<td>path statements not in corpus, and final</td>
<td>manual 83%</td>
<td>Fortran DEC PDP-9</td>
<td>Fill in the SHOMED blank, TMNTH right hand modifiers used in determinating code, new TMNTH started when repeat SHOMED category word occurs</td>
<td>690 entries</td>
<td>Errors: typo's 10%, not in dict 78%, wrong code in SHOMED statement 10%; dict constructed by hand, hard to change</td>
</tr>
</tbody>
</table>
sequentially using simple word match with logical operators. Search options included using the thesaurus to add search keys and thus increase recall. Stored narrative text consisted of surgical pathology, bone marrow, autopsy, nuclear medicine, and neuroradiology reports. A 1965 article by Lamson, Ginsky, Hawthorn, Soulter, and Russel described an earlier version of the same system which stored only surgical pathology reports. In both papers the working assumption was that exact syntactic or semantic differentiation was not required for retrieval. However, Lamson, et al., refute this assumption by devising a thesaurus that a trained interpreter must use to assist in the construction of queries. Although raw text was stored without syntactic or semantic differentiation, the burden of such differentiation was placed on the thesaurus and cleverly constructed queries. In addition, the 1975 article stated that updating the thesaurus was a complicated task. One must conclude that, although syntax and semantics complicate a system, they may not be ignored.

It is difficult to make generalizations concerning the methods used in encoding efforts described in Table 4-1. With a few exceptions, these methods center on constructing a lexicon (or dictionary). Dictionary size depends largely upon the scope and typicality of the corpus. Gordon (1970) provided an upper limit estimate of 24,000 terms found in medical publications for naming 3,700 to 3,800 specific diseases. He further estimated 10,000 synonyms or eponyms; 2,000 descriptors usually reserved for designation of manifestations; and 150,000 different descriptors, symbols, and abbreviations for signs, symptoms, and laboratory tests where 18,000 to 23,000 would suffice. These estimations are not beyond the scope of
Current computer technology but they do stress a need for careful storage allocation.

Other than dictionary use, the majority of systems in Table 4-1 defy categorization. Two methods, the "fruit-machine" and "fill-in-the-blank," deserve detailed explanation since they are reported by more than one group of researchers. The "fruit-machine" method, first reported by Howell and Loy (1968), uses a term dictionary. The dictionary entry for each term contains the codes for diagnostic statements which may possibly contain the term. Terms common to many diagnostic statements, such as "a," "the," "chronic," etc. are ignored unless they are needed in the code category. For example, using ICDA-8, "infarct" would have entries, 255.9, 410.9, 433.9, 434.9, 611.9, 444.2, 444.9, 615.9, 253.1, 573.9, 450, 634.9, 673.9, 631.0, 631.2, and 631.3. "Myocardial" would have 410., 410.0, 410.1, 410.9, 411., 411.0, 411.9, 412., 412.0 and 412.9. The code 410.9- "Acute myocardial infarction without mention of hypertension," would be selected for the statement, "Myocardial Infarction," since 410.9 is associated with both terms in the diagnostic statement. The term "fruit-machine," is descriptive of the method because a diagnostic code is selected on the basis of its appearing in the dictionary entry for each term in the diagnostic phrase; just as the same fruit or symbol must line-up in each category on a slot machine for a win to occur.

Howell's original "fruit-machine" design required "exact matching" of codes for all major terms. Thus if there were three major terms in the diagnostic statement, all three terms must have the same code in the dictionary. One non-matching term would prevent the "fruit-machine" from assigning a code. If a tie occurred, i.e., more than one code was associated
with each term, the machine consulted a "crossover dictionary," indexed by code number. One of the tied codes was selected if an "exact" phrase match was found in the "crossover dictionary."

Greenwood (1972), modified Howell's "fruit-machine" by requiring a "best fit" rather than exact fit. "Best fit" meant the code associated with the most terms was selected. He also devised a different method for resolving ties. The "preferred" code had an extra digit, which was equal to the number of terms in the diagnostic statement. The example Greenwood cited is of "diabetes," having 250., 253.9, 273.2, and 273.8 as possible codes. The 250 would have an extra digit (1), indicating that it was correct for the occurrence of "diabetes" without any other terms. Greenwood's "preferred" code method was motivated by Howell's report of consulting the "crossover dictionary" for 45% of all diagnostic statements. Greenwood's modification eliminated the need for an abundance of exact phrase storage.

The "fruit-machine's" success demonstrates the value of simple word co-occurrence. In fact, the machine has the power of a set of co-occurrence rules, such as, "Code 410.9, if 'myocardial' occurs with 'infarction.'"

Pratt (1973), and White, et al., (1977), reported using a "fill-in-the-blank" method for automatically coding into SNOP and SNOMED, respectively. Pratt explained the technique using the pathological diagnostic statement RETICULUM CELL SARCOMA, STOMACH as an example. First, STOMACH has an exact match with T6300-"Stomach, NOS." The remainder of the phrase is interactively searched for and permuted until M9643-"Sarcoma Reticulum Cell" is found. The algorithm's objective is to fill-in the
slots for Topography, Morphology, Etiology and Function (TMEF).

Pratt stated that the "fill-in-the-blank" method succeeded in coding 25% of his sample pathology statements. The remaining 75% needed some degree of syntactic or semantic disambiguation prior to applying the algorithm. Pratt improved the performance of his encoder by incorporating an algorithm to change adjectival forms to nominal forms. Thus, CORTICAL CYST would become CORTEX CYST, and LARYNGEAL BIOPSY became LARYNX BIOPSY. Nominalizing all words in SNOP and in the pathology statements provided a better chance for an exact phrase match between the two. The underlying motivation for these suffix changes was the fact that pathologists commonly use adjectival forms for brevity and readability.

Pratt stated that he believed his encoder, using SNOP, adequately captured the essence of pathology diagnostic statements. However, he never made an attempt to fully capture disease description, which involves "association, cause, effect, negation and qualification." In fact, Pratt's encoder is equivalent to an exact phrase match encoder with a very large store. The procedures of permutation and nominalization as well as the separation into TMEF categories are effective means of reducing storage.

There are possibly three additional unique encoding efforts reported in Table 4-1. The first is reported by Thatch (1964) for the U.S. Army. Thatch stated that for obvious reasons the Army has been interested in compiling morbidity and mortality statistics since 1818. He reported trying to devise an encoding system similar to a Godel numbering system for well-formed formulas. The idea was to assign a unique numeral to each word in a phrase, to look each word up in a dictionary for further
identification, and to form a sum which related to the desired code. The whole process was rather ambitious and would have been a promising achievement had it proven operable. Thatch reported success on a small corpus; however, lack of further publications on the method indicates that the idea was dropped.

An effort conducted by Sager, et al., (1976), at New York University as an extension of the Linguistic String Parser research, used their parser to identify (verb, subject) and (verb, object) pairs. They then clustered subjects and objects used with identical verbs to form word classes. The word classes thus became format slots which were filled during encoding. This technique allowed for structuring the storage of multi-sentenced text. The purpose was to derive format slots for each specialty from typical texts, enabling storage of complete information in a structured form. Thus far, only a small sample of radiology reports and an article on dialysis have been structured. It seems that automating construction of word classes caused some problems. The whole process must bear two major criticisms when applied to medical text. First of all, many medical statements are not complete sentences but phrases. Often no verb is even implied other than possibly forms of "to be." Selection of (verb, object) and (verb, subject) fails without a verb. Secondly, derived format slots only represent surface structure. In this respect, the technique seems little better than keyword search of free text.

The final technique to be discussed was reported by Mishelevich (1972). He built a semantic tree called a "sparse" which was similar to a semantic net. Mishelevich characterized problem-definitions as noun phrases and used the noun in each phrase as the root of the tree.
Modifiers formed branches. He proposed to attach logical connectors to the trees but he did not indicate how such a procedure would be automated. No testing of the method was reported and the basic idea has not been further developed.

One may conclude from Table 4-1 that much energy has been expended over a long period of time to attempt to computerize Natural Language medical text; and that active interest still exists for achieving an acceptable coding process. Although correct coding can be above 90% in prototype systems, an optimal meshing of linguistics, artificial intelligence, and medical reasoning has yet to be achieved. In other words, much work remains to be done.

**Natural Language Processing's Role in Automated Medical Encoding**

This section outlines the history of Natural Language processing research and summarizes the contribution of various researchers within the field of Natural Language processing. The following list contains an overview of results which have some consensus of agreement within the field of Natural Language processing. Individual contributors to the list are documented in succeeding paragraphs. The list is not exhaustive; but it gives direction which anyone beginning work on Natural Language processing might glean from past work.

1. Syntax (surface structure) alone is not sufficient to find even parts-of-speech of words in a sentence (Yngve, 1964).
2. A successful parser must somehow make predictions about
what it will encounter (Kuno, 1965).

(3) Natural Language communication with a machine can have some success when the domain of discourse is limited (Winograd, 1972 and Woods, 1970).

(4) A synonym dictionary does not greatly improve text understanding, nor does a thesaurus (Simmons, 1972).

(5) Text understanding involves integration of knowledge about syntax (form), semantics (word sense meaning), and real world knowledge (Schank and Colby, 1973 and Winograd, 1972).

(6) Machines which "understand" must have inference mechanisms (Schank and Reiger, 1973).

When computer development was still in its infancy, linguists primarily, recognized the computer's ability for symbol manipulation. Projects were funded for machine translation. This was in the late 50's and early 60's when the "cold war" necessitated translation of much Soviet block literature. These early attempts at machine translation have been considered a dismal failure—especially in view of initial projections. There is general agreement that oversimplification of the problem was a major source of failure; but this recognition, in itself, is a contribution to Natural Language processing.

Chomsky's (1965) transformational grammar theory greatly influenced early work. Emphasis was on identifying syntactic structure and many workers used grammars as the basis for their syntactic parsers. Yngve (1967) wrote a context free grammar from ten sentences in a children's
story. He used this grammar to generate sentences. Of course the sentences were grammatically correct but meaningless. Conclusions were that knowledge of a grammar which produced syntactically correct sentences was not enough. Another researcher, Kuno (1965) showed that prediction was useful in parsing sentences. He was among the first to recognize practical implications of language ambiguity. His parser became more "intelligent" by expecting, for example, a noun or adjective after encountering an adjective. Other researchers at this time came to the same conclusions. Therefore items 1 and 2 on the preceding list are the primary contributions of machine translation work.

Anderson (1976) reported that Winograd (1974) in a series of lectures distinguished two generations of interactive language understanding systems. Winograd characterized the first generation's goal as trying to give the appearance of understanding within a limited task domain. Weizenbaum's (1966) ELIZA and Colby (1973) PERRY are typical of this generation. ELIZA simulated a psychiatrist and PERRY simulated a paranoid patient. These programs primarily looked for keywords or phrases and parroted standard replies from their repertoire. Sometimes the dialogues were appropriate and sometimes they were not.

Other first generation programs were more ambitious but they also restricted their task domain. Green, Wolf, Chomsky, and Laughery (1963) designed BASEBALL to answer questions about baseball. They used, what AI would call, a set of question templates with empty slots. When a question was posed, the slots were filled. Those slots which remained empty were searched for in the data base as the answer to the question.

Lindsay (1963) developed a program, SAD-SAM, to answer questions
about kinship relations. Lindsay used a left-to-right parser which he said was psychologically logical. He also used a finite intermediate memory to limit recursive depth and an associatively organized memory. A word dictionary aided the parser by providing parts-of-speech. The semantic analysis was separate from the syntactic part-of-speech parser. It relied on a structure similar to a family tree.

Raphael's (1968) Semantic Information Retrieval (SIR) system was capable of storing and retrieving information dealing with set relations, part-whole relations, ownership, and certain spatial relations. The system deemphasized parsing by matching input sentences against a set of templates. Output was accomplished in a similar manner. Memory was organized along set theoretic lines and built from "conversations" with humans.

These first generation programs had limited domains but they were written as ad hoc games with no claim to "understanding" Natural Language. Therefore, the first generation of researchers in Natural Language processing provided a prelude to item 3 on the previous list.

Winograd characterized second generation efforts as emphasizing mechanism development and process correctness rather than result appearance. If one were forced to put dates on this generation, the late 60's and early 70's would probably suffice. In this era Quillian (1969) proposed semantic nets. Woods (1970) formalized augmented transition networks, and Schank (1972) put forth his Conceptual Dependency theory. These works all emphasized knowledge representation.

Chomsky's (1965) deep structures and Fillmore's (1968) deep cases greatly influenced second generation Natural Language processing.
researchers. It was generally concluded that machine understanding re­quired a knowledge store far removed from the surface structure of langu­age. Simmons' (1972) stated the emerging concensus, "In my opinion, a semantic representation should probably be completely free of its Natural Language representation." Item 4 on the list was the result of thinking within this second generation because these researchers showed that simply storing words or finding correct word sense meanings was not enough.

Current research interests blend with the second generation be­cause verdicts as to how to represent knowledge are not in. Several re­searchers, Simmons (1972), Quillian (1968), Charniak (1977), favor networks. Even Minsky's (1975) frames and Schank's (1977) scripts may be viewed as networks. A major question is whether to place meaning on links within the network. Simmons (1972) explicitly prefers to only attach meaning to nodes. Other workers opt for labeled links. No one has shown which is best or works better.

Another current open question is whether to represent knowledge as procedures or structures. Charniak (1975) gave a good discussion of the pros and cons. Winograd (1972) spearheaded the procedural view. Ander­son (1976) modified this view and attempted to give psychological evidence for its use. Schank (1977) and Minsky (1975) are chief proponents of a structural representation.

The latest issue is how to use inferences in Natural Language pro­cessing. The necessity for their use is undisputed. However, when to use them (at read time or question time) and how to store them (as demons or structures) is still undecided. In fact, there are many unanswered
questions. The following list indicates a few.

(1) What is understanding and how can it be measured?

(2) What is intelligence and how can it be measured?

(3) How should knowledge be represented for optimal machine "understanding?"

(4) How do syntax, semantics, and pragmatics interact in understanding?

(5) How can a program be made to select the knowledge which it uses in interpretation and how can it be made to ignore other information?

(6) When should inferences be made—at input time or output time?

(7) How may the enormous amount of knowledge recognized as needed for machine processing of Natural Language be stored and retrieved (serial or parallel search)?

Automatic encoding of problem-definitions is applied Natural Language Processing research with a restricted domain of discourse and a restricted surface structure. Problem-definitions are not complete sentences, nor are they organized into paragraphs as one finds in many written texts. They are noun phrases with an understood subject, the patient. Hence, it is not necessary to have a comprehensive syntactic analyzer of English or a general parsing algorithm and only limited inference mechanisms are needed. Chapter VIII describes the Natural Language processing required to encode problem-definitions.

Future research in Natural Language processing with medical language
may develop a medical knowledge structure similar to the type of structures currently envisioned by researchers in Natural Language processing. However, there are important questions of what constitutes a deep, conceptual representation in medicine and whether such a representation is useful in a practical coding scheme, e.g., how should a diagnosis be stored, as a primitive unit or by its manifestations. This is a very basic question with no satisfactory answer. Some sort of script or frame structure seems a plausible solution. With this scheme a diagnosis would have a set of simple clinical manifestations and an explicit representation of the progression from one health state with an associated set of characteristics to another. One could then infer that by stating a diagnosis, the physician was stating that the associated set of clinical manifestations also occurred. Indeed, this is one use for a diagnosis, it is a name of a set of signs and symptoms which characterize a disease.

But there is difficulty with taking this approach at this time. The major difficulty is that medicine has not explicitly defined the set of signs and symptoms associated with each diagnosis. Of course there are reference books which index differential diagnoses, but these lack the precision necessary to implement a script or frame. The major deficiency is the absence of probabilities to qualify signs and symptoms. No physician would accept vomiting as a symptom whenever appendicitis was displayed from a computer. However, he might accept a display listing vomiting as occurring in 60% of all cases of appendicitis.

The current research develops a method to easily collect large amounts of medical data and thus provide the ability to calculate the necessary probabilities for a complete medical knowledge structure. In
addition, it incorporates limited inference mechanisms which are necessary
to map from Natural Language statements, which often contain redundant,
ambiguous, and incomplete information, into a semantic representation
suitable for machine processing.
CHAPTER V

RESOURCES

Design Set Problem-Definitions

This investigation uses a set of problem-definitions collected at Community Health Care Plan, Incorporated (CHCP) to design an automatic encoder. CHCP is a Health Maintenance Organization (HMO) in New Haven, Connecticut. From March 13, 1973, through November 1974, CHCP recorded enrollment identification information, reason for visit, vital signs, normal physical findings, problem-definitions, medications, signs, and symptoms on a sixteen page Clinical Visit Form (CVF). This form was developed by CHCP and the Medical Computer Sciences section (MCS) of Yale University of Medicine as part of a computerized medical record system.

While CVF was in use, it was the only record kept for outpatients. The practitioner received the outpatient's chart and a CVF form with pre-printed identification information prior to each patient's visit. Either during or immediately after the visit, the physician recorded pertinent visit information on the CVF. CVF data was then keypunched and stored at MCS using a PDP-11/20. The computer generated a "single visit report" and a "cumulative report" which were placed in the patient's paper chart as a record of the visit. During the period when CVF was in use, over 140,000 visits were made by approximately 16,000 patients.

Some inpatient data was recorded on the CVF's. Lynch (1975)
reports that most CHCP hospitalizations occurred at Yale New Haven Hospital. From September 1973 through January 1975, a nurse monitor visited Yale New Haven Hospital wards daily and abstracted CHCP enrollee information onto CVF's. This inpatient data was then processed in the same manner as the outpatient CVF. Approximately 478 inpatient CVF's were computer processed.

All the CVF computerized data was available on the Biomedical PDP-11/45 computer at the Georgia Institute of Technology. Dr. S. Brunjes brought both computer and data to Georgia when he became Director of Biomedical Information and Computer Science at the Georgia Institute of Technology. The remainder of this section explains the appropriateness of using the CHCP problem-definitions as a design set.

Any extrapolation from CHCP problem-definitions to a general population of problem-definitions must consider the CHCP setting. CHCP enrollment came mainly from employment groups which selected CHCP services in lieu of other health insurance plans. The few fee-for-service patients generally were covered by MEDICARE and the Public Health Service. During the time CVF's were processed, fee-for-service patients accounted for approximately 11% of the CHCP enrollment file and 5% of services provided by CHCP.

The cost of CHCP membership restricted the population using CHCP. Most enrollees were from a white, middle-class, working population. Therefore, CHCP problem-definitions are not representative of the types of problem-definitions one might find in more heterogeneous populations. In fact, only a small fraction of the CHCP subscribers were over 65 years old and a disproportionately large fraction of enrollees were women aged
22-30 years (Lynch, 1975).

Although the CHCP problem-definitions are not representative of all problem-definitions, there are several reasons for their use in this investigation. First, CHCP problem-definitions were available in computerized form, unbiased by any other Natural Language processing. For economy, it is reasonable to try to gain as much information as possible from what data exists before spending more money to collect new data.

Secondly, there is a large number of CHCP problem-definitions (127,606), collected over a period spanning more than a calendar year. Since problem-definitions vary with the seasons (flu more prevalent in winter and bee stings more frequent in summer), the length of sampling time is important. However, despite this fact, the CHCP problem-definitions are not a random sample. During the time in which CVF's were computer processed, CHCP enrollment increased steadily from 8,459 to 14,391 so there are a larger number of problem-definitions from the final months in which CVF was used. Outpatient problem-definitions from spring, summer, and fall months are represented twice in the sample while December, January, and February problem-definitions occur only once. The few inpatient problem-definitions have the months of September through January represented twice. One would expect to detect a disproportionately low number of winter related problem-definitions from outpatients and also a low number of winter, spring, and fall problem-definitions from inpatients. However, the objective of the present work is not to analyze the occurrences of certain types of problems but rather to automatically encode a variety of medical problems stated in many forms. For this purpose, the large number of problem-definitions collected over at least a full year
seems adequate and appropriate.

The CHCP problem-definitions do not represent a random sample for another reason. CVF was modified twice during the time when the computer captured data (August 1973 and May 1974). Its last revision contained eight preprinted problem-definitions. These were included in the form because they numbered among the most frequent problems encountered at CHCP and physicians saved time by checking these standard problem-definitions instead of having to write them out. Without further information, it is impossible to ascertain whether the preprinted problem-definitions biased physician's recording of problem-definitions. This research simply rests on the assumption that the preprinted problem-definitions may occur more often than other problem-definitions because they were more convenient to check.

A more substantive bias was introduced into the comment portion of CHCP problem-definitions. There are only eight different comments in the whole list of problem-definitions. These comments correspond in part to those preprinted on the CVF. CHCP physicians were probably influenced by the preprinted comments so no generalizations about comment form can be attempted from the CHCP problem-definitions.

Lynch (1975) mentioned one final bias introduced into the computerized CHCP problem-definitions by keypunchers. He stated that problem-definition fields were restricted to 40 characters and keypunchers sometimes abbreviated lengthy problem-definitions. Clearly, some abbreviations at CHCP were not commonly used medical abbreviations. Therefore, any conclusions about medical abbreviations must be tempered by keypunching bias.
Even though there are imperfections in the CHCP sample of problem-definitions, two further arguments add to the value of the sample. First, a completely random sample of problem-definitions is cost prohibitive. No single institution could provide a random sample from the total population of problem-definitions and the cost of sampling from many institutions could not be justified for study of automatic encoding since no workable automatic encoder of problem-definitions exists to process the variation of syntactic and semantic form within even one institution. Initial efforts must show that the information in problem-definitions from one institution can indeed be automatically encoded.

Finally, data other than problem-definitions is available in computerized form for the CHCP patients. An important addition to the present work will be the future correlation of signs and symptoms, coded in the same coding scheme as problem-definitions, with problem-definitions on the same patient. Although the present work intends only to code the problem-definitions, much more information can be gleaned from data contained in the CVF's.

**Past Work Using the Design Set of Problem-Definitions**

Past work with CVF data is a resource utilized in the present study. Lynch (1975) wrote a master's thesis using the CVF data for the Department of Epidemiology and Public Health at Yale University. His thesis is the primary source of information about the CHCP setting. His objective was to quantify HMO use by variables such as age, sex, family position, family size, and employment group and relate these to problem, sign, and symptom. HMO administrators must be able to predict resource
usage from measurable attributes of their subscribers. Since some estimate may be made on cost of care for certain types of problems, Lynch's work provides the necessary link between population and problem.

Powsner (1978) also used the CHCP problem-definitions in his thesis. He designed an automatic encoder and coded approximately 1,000 of the most frequent CHCP problem-definitions into ACS System and Function dimension codes. Powsner's encoding method proceeded generally as follows. Each word in a problem-definition was looked up in a dictionary. Dictionary entries indicated either: (1) ignoring the word, (2) retrieving the dictionary entry of the word correctly spelled, (3) specifying a System or Function code, (4) specifying the type of word, modifier or etiology, or (5) specifying forward and backward linking with the other words in the problem-definition to form a phrase. As each word was found in the dictionary more than one code might become possible for each problem-definition because some words were multiply defined. Powsner chose always to select the code with the fewest number of System and Function interpretations as the best. He also provided a special subroutine to handle coding ambiguous abbreviations and words such as INF and COLD.

Test Set Problem-Definitions

A test set of problem-definitions was available from the MARIS project at the Georgia Institute of Technology (Slameka, Camp, and Badre, 1977). This set was used to test the general applicability of automatic encoding mechanisms designed here. A brief review of three sample patient records indicated that there was some similarity between the CHCP problem-definitions and those MARIS was collecting from Grady Memorial Hospital in
Atlanta, Georgia.

The MARIS staff, with permission from Grady Hospital, supplied 100 of the most frequent problem-definitions collected from January 1978 to June 1978. Appendix A contains a listing of these problem-definitions. The test set problem-definitions are from a self-care ward in a large, inner-city hospital. The demographic characteristics of the test set patients and the design set patients are totally different. The CHCP population is generally white working class with employer-supported medical insurance. The Grady population is generally black with medical care paid by government agencies.

The MARIS project employs a physician's assistant to input data from the paper medical record into the computer. Her instructions are to transcribe the data exactly as it appears in the paper medical record. However, a subconscious review of the data does take place in transcription. The physician's assistant has access to the medical personnel originating the data so she may ask them for additional information or clarification. Therefore, one would expect the computerized data to be in slightly better form than the original, unreviewed medical record.

The problem list acceptable at Grady allows a free style progression of problems over time. A string of arrows (→) indicates a change in the problem. Often these arrows are labeled with dates. In addition, some problem-definitions have subcategories of comments or explanatives listed beneath the main problem statement. Retrieval of the test set requires removing the arrows and denoting the information between the arrows as a problem-definition. In addition, all dates are changed to one fictitious date to protect patient confidentiality. Since all dates
are the same, the test sample contains a few infrequent problems.

The test set was selected from 225 patient records collected over a six month period. During part of this time the MARIS staff were developing the system; therefore, not all the ward's patient records were input into the computer. The ward has 30 beds with an average weekly turn-over. If all patient records from the ward were input into the MARIS system, the 225 total patient records would represent a 10 week period.

**Equipment**

The School of Information and Computer Science's Biomedical computer, a PDP-11/45, was used for implementation of the automatic encoder. Convenience was not the only consideration influencing the choice of machines—potential cost effectiveness was also a criterion. The PDP-11/45, being a medium sized minicomputer, would be cost effective for implementing an automatic encoder in a clinical environment. In addition, Powsner's subroutines and dictionary were available on this machine.

UNIX, developed and supported by Bell Laboratories, was the operating system in use on the Biomedical PDP-11/45. It offers the user versatility and power. Many file manipulation tasks do not require programs because they are available as commands to the UNIX operating system. Four programming languages are supported under UNIX, SNOBOL, C, Fortran, and Basic. C was chosen for the automatic encoder for several reasons. One important reason was that the other languages available under UNIX had characteristics which made them less appropriate for use in this research than C. Users at Georgia Tech having experience with
the UNIX variant of ANSI Fortran report bugs in the compiler. SNOBOL is a string manipulation language primarily used for pattern matching. Basic is an interpretive language useful for teaching programming to beginners and some scientific applications. C is the structured language in which most of the UNIX operating system is written; therefore, it is versatile and the compiler is reliable. It is not as portable as Fortran with the RATFOR preprocessor; but as one user states, it has "deceptive power through simplicity."
CHAPTER VI

METHODOLOGY

Prepare Design Set

As described in the previous chapter, the CHCP problem-definitions exhibit the variety of semantic content and syntactic form used by physicians when they record a patient's medical problem. At the onset of this investigation, a decision was made to preprocess the problem-definitions to reduce any surface structure variation which was either introduced into the data by keypunching or was such as could be reduced algorithmically. The reasons for attempting to reduce these surface structure variations by preprocessing were:

1. Types of variation introduced by keypunching are not likely to be a good sample of the variety which will be inserted in a real-time environment.

2. Algorithmic preprocessing of problem-definitions is easily achieved and is peripheral to the objectives of this research.

3. The time to accomplish succeeding steps is reduced because only one copy of each problem-definition which occurred more than once needs to be manually encoded.

There are 29,003 unique problem-definitions in the original 127,606 CHCP
problem-definitions. The phrase, "unique problem-definitions" will be used to denote the minimum set of problem-definitions with unique character strings.

Spelling correction is one preprocessing task which was performed in as expeditious a manner as possible because many spelling errors were obviously the result of keypunching errors, keypuncher interpretation, or physician misspelling. These sources of variation are either peripheral to or not appropriate for the intent of this research.

Spelling error detection and correction is a valuable research area but it has a fairly extended history of investigations. Alberga (1967) reviewed and compared eight different methods for spelling error detection. Spelling correction methods rely mainly on selecting a correctly spelled word from a dictionary after an error has been detected. In addition, detection and correction of misspellings does not have the importance today that it had before the advent of real-time processing. Current acceptable methods simply ask a user to either correctly spell a word or add it to a dictionary.

This investigation used the 4,858 spelling corrections in the Powsner dictionary and added an additional 673 spelling corrections in the current word dictionary. Most of the additions were made from a word frequency list of the original problem-definitions; however, a few were made after the word was viewed in context. The spelling correction entries remain in the current word dictionary and provide a link back to the original set of CHCP problem-definitions; however, during preparation of the design set, a substitution of the correct work for the misspelled word was made to reduce the number of unique problem-definitions.
Differences in the forms of abbreviations in the original CHCP problem-definitions were a source of surface structure variation which could be algorithmically reduced. For example, "B.P", "B P", and "BP" were all in the CHCP problem-definitions as abbreviations for blood pressure. All abbreviations in the design set were normalized by a program which removed blanks and periods between a series of more than one alphabetic character separated by blanks and/or periods. The subroutine which normalizes abbreviations is included in the preprocessing section of the automatic encoder in order to retain a link back to the original CHCP problem-definitions.

To further reduce the number of unique problem-definitions in the CHCP set, another program removed standard comments appearing at the end of many problem-definition strings. The Clinical Visit Form had pre-printed comments which were included in many problem-definitions. Some were separated by semicolons from the rest of the text and others had no separators. Figure 6-1 lists the eight standard comments removed from the original problem-definitions. These comments provide no challenge to an automatic encoder because a simple one-to-one mapping easily encodes them. However, they do make problem-definitions such as OBESITY, OBESITY NEW, OBESITY NO CHG, OBESITY WORSE, etc., unique. Again, a link is retained to the original CHCP problem definitions since the subroutine which removes the comments is included in the preprocessor section of the automatic encoder. In addition, the automatic encoder will assign dimensional codes to any comments which are removed during preprocessing.

One final method was used to reduce the number of unique problem-definitions. All LEFT, L, LT, LFT, RIGHT, and RT's were changed to R.
WORSE
UNDER CONTROL
SEE DICTATION
RESOLVED
NEW
NO CHANGE
IMPROVED
SYMPTOMATIC

Figure 6-1. List of Standard Comments in CHCP Problem Definitions
In only one instance, that of the heart, does the encoding of left differ from that of right. Therefore, no generality is lost by designing the encoder only on the word "right." Moreover, all codes for "left" are available to the automatic encoder; but none are used in the design set. The final number of unique problem-definitions was 17,364. More than 90% of the original problem-definitions, i.e., 114,121 out of 127,606, occurred more than once after each problem-definition was modified in the manner described above. The 5,465 unique problem-definitions occurring more than once became the design set for the automatic encoder.

One additional mechanism which could have been used in the automatic encoder is word stemming. Word stemming can reduce dictionary size since a word dictionary need contain only the word stems from many medical English words which are composed of meaningful stems. Moreover, the addition of a stemming algorithm to a dictionary look-up routine allows that routine to possibly recognize and construct definitions for words which have not been previously encountered. Word stemming was not used because it is a peripheral issue in this research and it is a current object of investigation by Pacak and Pratt (1978) at the National Institute of Health.

**Create New Dimensions**

The next step was to extend ACS to reflect all the concepts within the design set problem-definitions. In this step a word frequency list of all words in the original CHCP problem-definitions was used to make a preliminary assignment of dimension types and/or codes to words. In addition, the list contained the definitions of any words which were in the
Powsner dictionary. For this research a word is "defined" by a dictionary "entry" which contains dimensional codes for the concepts conveyed by the word. Chapter VIII describes the form of an "entry" in the current word dictionary.

There were several passes through the word frequency list. In the first pass codes were assigned to words containing System and Function concepts. This was possible during the first pass because these dimensions had been structured by Powsner (1978) and Lynch (1975). Codes were added to System and Function when needed. No codes were deleted since the signs and symptoms on the CVF form for these CHCP patients were previously encoded into System and Function and future work may associate signs and symptoms with problems coded in the same coding scheme.

During the first pass the types of new dimensions were also assigned to some words. For example, we already knew that some dimensions must contain time concepts; therefore, words which conveyed some meaning of time were marked. Other words representing concepts such as medications, people other than the patient, and common activities which people engage in were also distinguished. The objective in this pass was to identify classes of concepts which might exhibit similar uses in problem definitions. The pattern recognition ability of a medically knowledgeable human was the primary mechanism used to form concept classes which evolved into dimensions.

During the first pass many words were left undefined, i.e., without codes assigned; in succeeding passes the additional dimensions were structured and more words were defined with codes. Between passes through the entire word frequency list, words which had the same markings were listed
so that the similarities and differences among the concepts they conveyed could be subjectively judged.

Three major constraints were applied during the construction of dimensions:

(1) The dimensional structure must facilitate aggregate retrieval.

(2) The dimensional codes assigned to a problem-definition must contain all explicit information within the Natural Language statement of the problem.

(3) An automatic encoder must be able to have some criteria for acceptance or rejection of a set of dimensional codes.

The first constraint is essential for justification of an automatic medical record system. The major reason for automation is the ability to quickly and inexpensively retrieve and manipulate medical data. This investigation does not evaluate the retrievability of medical data coded into ACS. Finseth, Dallal, Freeman, and Brunjes (1977) have shown that ACS is useful for aggregate retrieval of patient signs and symptoms. Since ACS theoretically has the potential for coding all information in the medical record, a thorough retrieval evaluation must await future results.

The significance of possibly representing all information within the medical record with one coding scheme must not be underestimated. No other medical coding scheme has such a potential. This research is one module of a plan to eventually construct an automatic medical record system using ACS. First signs and symptoms were coded, and this research
codes problem-definitions. Other current efforts are directed at coding patient history and developing a data management system. The full import of ACS cannot be evaluated until additional modules are complete.

A coding scheme must meet the second constraint to be useful in everyday clinical care, i.e., all the information in a problem-definition must be able to be reconstructed (decoded) from the dimensional codes. Successful achievement of this means that duplicate forms, coded and Natural Language, do not need to be stored for a problem-definition.

The third constraint allows an automatic encoder to perform a more sophisticated task than merely lexical retrieval. A lexical retrieval automatic encoder simply displays the dictionary entries and all decisions are made by a user; it is a substitute for a coding manual. If the automatic encoder has some means of evaluating the codes it produces, then it can select the most appropriate coding. Final responsibility for determining coding correctness rests upon the user; however, a sophisticated automatic encoder is instrumental in decreasing the amount of time the user needs to make a correctness decision.

**Design and Implement Automatic Encoder**

The next step was the design and implementation of the automatic encoder. The only constraints at this time were that the encoder program must be versatile and allow the user to modify the word dictionary. It must also produce ACS codes which agree with those assigned by a manual encoder. The next section discusses the evaluation of codes produced by the automatic encoder.
Evaluate the Automatic Encoder's Ability to Produce Codes

Agreeing with a Manual Encoder

A diagnostic test in medicine is often evaluated on the basis of its percentage of false-positives and false-negatives. A false-positive occurs whenever a test indicates something amiss with the patient; when in reality, he does not have the dysfunction which the test was designed to detect. A false-negative occurs whenever the test misses the dysfunction and evaluates the patient as normal when, in reality, the patient has the specific dysfunction. In medicine, a false-positive is usually a less serious error than a false-negative because other tests may be available to screen the patient before actual treatment begins. Of course the risks involved in testing the treatment may alter this statement. Statistically, a false-positive is a Type I error, and a false-negative, a Type II error. An evaluation criterion analogous to that used in medicine for diagnostic tests was used for the automatic encoder.

Often in evaluating medical diagnostic tests, reality is subjective, i.e., some authority judges whether a diagnostic test correctly portrays reality. Although this situation is less than optimal for determining the worth of a diagnostic test, an alternative, such as waiting for the patient to die and then conducting an autopsy to gain objective judgment, is generally not acceptable.

Applying subjective judgment as to whether an automatic encoder correctly encodes a problem-definition is clearly not as critical as deciding whether a diagnostic test correctly detects an illness in a patient. However, the difficulty in obtaining objective judgment for an automatic encoder outweighs the expected small error for using
subjective judgment. This is especially true since the implications of an automatic encoder can be projected far into the future.

An automatic encoder must be evaluated ultimately on retrievability within the setting of a completely automated medical record. Since the encoder and such an environment are interdependent, this investigation is a first step toward achieving an automated medical record system with the ability to accept Natural Language input. Final evaluation must await the completion of many other steps such as a database management system.

The preceding paragraphs preface discussion of the method used to evaluate the automatic encoder implemented in this research, since a completely objective evaluation is impossible at this time, i.e., there is no existent automated medical record system using ACS. In addition, the value of an additional subjective judgment, such as that obtained from a panel of physicians, does not outweigh the expense of conducting such an evaluation. This type of experiment would involve not only the time for a group of physicians to assign a code to the problem statements, but also the time to thoroughly indoctrinate them in ACS.

Determination of correct manual codes was made by this researcher, with professional medical consultation. At the current stage in the development of an automated medical record system using ACS, this is not a weak or inadequate procedure for several reasons. First, the objective was to design both a coding scheme and an automatic encoder. The person coordinating the design of both the coding scheme and the automatic encoder had to be thoroughly versed in both. Secondly, repeatedly encoding the design set and comparing the automatically assigned codes to those manually assigned ensured consistency, i.e., if a change were made to fit
a specific problem-definition, the ramifications of that change became
evident after the next iteration. Thirdly, many of the problem-definitions
were simple statements; CHEST PAIN is an example which required no detailed
medical knowledge. Finally, this research was performed in consultation
with a physician.

For purposes of evaluation, there are two global perspectives--
that of the automatic encoder and that of the manual encoder. The manual
encoder’s perspective is reality and the evaluation is conducted to con­
firm that the automatic encoder’s perspective is identical to the manual
encoder’s view. The automatic encoder must be able to detect two situa­
tions:

(1) It must correctly judge when a statement is codeable.

(2) It must detect those statements which a manual encoder
considers not codeable.

A failure in the first case is a false-positive; and, in the second case, a
false-negative. Since the intended environment for the encoder is
real-time, a false-positive is a less serious error than a false-negative
because manual encoding can override any automatically assigned codes.

One additional feature enhances the use of an automatic encoder--
the ability to ask a user for information needed to code a statement.
In other words, if the automatic encoder can state the reason for its
inability to encode a problem-definition, it can ask the user for speci­
fic information. This ability to interact with a user was designed into
the automatic encoder and its implementation is discussed in Chapter VIII.
Evaluate the General Applicability of the Automatic Encoder

With any finite set of data, a simple automatic encoder can produce codes in perfect agreement with manual codes. Achieving this requires only a table look-up. The objective of this investigation was to design and implement an encoder which would correctly encode problem-definitions included in the design set as well as problem-definitions which the encoder had never processed. To test the generality of the automatic encoding techniques designed and implemented in this research, the automatic encoder was run using a test set of problem-definitions from a source other than CHCP.

Manual codes were assigned to the test set of problem-definitions using the automatic encoder in the same manner as was used with the design set. However, no modifications were made to the automatic encoder since this test was to determine how the automatic encoder, as tuned, performed on a totally foreign set of problem-definitions.
CHAPTER VII

DESCRIPTION OF THE ANAMNESTIC CODING SCHEME

Meeting Criteria for Dimensions

ACS was extended from the original System and Function dimensions under the three constraints listed in Chapter VI. These constraints placed different perspectives on the task. In some instances there were compromises; but the overall effort attempted to satisfy the requirements of each constraint.

Specific numeric codes within the System and Function dimensions were determined prior to the current research. This research retained the numeric arrangement within System and Function and constructed similar codes for new dimensions. Each dimension uses 16 bits for a numeric code. There are four hexadecimal digits allowing for a depth of four in the hierarchy. A retrieval of all heart problems, for example, involves keying on the first two hexadecimal digits (52) in the System dimension. Any specific subsystem within the heart is designated by the two, lower order, hexadecimal digits.

Some decisions necessary for proper construction of a hierarchical order are difficult to make. For example, is blood pressure to be subordinated under the Cardiovascular System or under the Hemic, Reticuloendothelial System? In the System dimension, there is some precedent for a hierarchy; all diagnostic coding schemes in current use have some arrangement by System. In other dimensions, which have no correlates
with any existing coding scheme, the hierarchical arrangement is a totally original approach.

Retrieval using ACS also involves the concept of a coded clinical event. Each clinical event is descriptive of some incident pertaining to the patient. In fact, the entire medical record is composed of clinical events. ACS provides a way to store the information within a clinical event in a form suitable for computer manipulation. Each clinical event, when correctly coded into the dimensions of ACS, is unique.

To maintain uniqueness, each coded clinical event may have each dimension represented only once, e.g., there never are two System codes within one coded clinical event. Indeed, in any coordinate system, there is only one value for each axis. In most instances, this constraint applies readily; however, it presents some difficulties for coding a few problem-definitions. For example, a fistula occurs between any two adjacent body cavities or organs. If the two cavities or organs were coded into one System dimension, System would have to contain a list of all pairs of adjacent cavities or organs, and it would be impossible to observe any hierarchical ordering. Furthermore, it would destroy the ability to retrieve, with one key, all problems dealing with one of the organs or cavities. A solution is to code two separate coded clinical events, one with each organ or cavity, and put a linking dimension code, Continuation, in one of the coded clinical events. Therefore, COLOVAGINAL FISTULA codes as (CSY6460-"Large Bowel," CFU9A70-"Fistula") and (SCO0100-"Continuation of Conjoined Problems," CSY7740-"Vagina," CFU9A70-"Fistula"). The abbreviations, CFU, CSY, and SCO are explained in the next two sections of this chapter.
There are a few other situations in which the Continuation dimension is used; however, these are discussed later in this chapter. The point being emphasized here is that each coded clinical event contains not more than one code per dimension. Furthermore, Function is the only dimension required to appear in each coded clinical event. The other dimensions are implicitly assumed to be null when they do not appear in a coded clinical event.

Specification of what dimensions constitute a coded clinical event provides an automatic encoder with some criteria to determine the correctness of a set of dimensional codes, i.e., a correct coded clinical event must have a Function code and no dimension may be represented more than once. These are the only correctness criteria applied in the current investigation; however, two categories of additional criteria could be applied.

The first category would be imposed by an institution to ensure completeness. One criterion in this category might be to require a System code. For example, one CHCP problem-definition is CYST which produces a correct Function code but there is no indication of the cyst's location. If the automatic encoder were required to assign a System code to every coded problem-definition, then it would ask the user to indicate the location of the cyst.

The second correctness category requires inference capability on information from the rest of the medical record. For example, an automatic encoder with this ability would check the patient's sex whenever HYSTERECTOMY is encountered. Criteria within this category were not implemented in the automatic encoder since they are viewed as a separate,
future project extending the current work.

The second constraint listed in Chapter VI specifies that ACS must contain all information within a problem-definition. The current research did not attempt to code all implicit information in Natural Language problem-definitions since this necessitates a complete medical knowledge structure. However, during the extension of ACS an attempt was made to capture all explicit information in Natural Language problem-definitions within the ACS dimensional codes. One way to evaluate success under this constraint would be to design a decoder which translates from ACS code to Natural Language. The current research does not do this; however, it does provide logical links among dimensions which make decoding theoretically possible. The final section in this chapter summarizes the logical relations among dimensions for the clinical Event Type problem-definition and postulates relations among ACS dimensions for another Event Type.

Finally, the hierarchical arrangement of dimensions within ACS enables the automatic encoder to eliminate some redundancies in the Natural Language problem statement. This benefit of ACS was not initially perceived; but it was useful in encoding many problem-definitions. For example, in CONJUNCTIVITIS R EYE, conjunctivitis immediately designates the System as the conjunctivae of the eye. But, the next two words also specify a System—the eye. Since CSY2500-"Conjunctivae" is subordinate within the hierarchy to CSY2000-"Eye," the automatic encoder easily detects redundant information, and codes only the most specific System code. There are many such redundancies in the problem-definitions. Indeed, redundancies are necessary in Natural Language for error checking. The
ACS hierarchy provided the automatic encoder with a way to eliminate redundancy in assigned codes. It also utilized the redundancies for error checking in a manner similar to the way it is used in Natural Language, i.e., if two codes from the same dimension are not related by subordination within the hierarchy, then a possible error occurs.

**Groups of Dimensions**

The dimensions may be grouped according to their importance in clinical care and aggregate statistics. The names of the groupings are: Core, Identifier, Descriptor, Modifier, and Specifics. These groupings are arranged primarily for purposes of storage and retrieval within a data base management system. For example, most aggregate statistics will probably need only the information in the Core and Identifier dimensions for initial retrieval. Storage will use the information in the one Specifics dimension, and decoding for clinical care will use all the dimensions. The following paragraphs in this section discuss the groupings in further detail and Figure 7-1 may be used as a reference for the remainder of this Chapter.

**Core Dimensions**

Four dimensions comprise the Core dimensions which contain the main concepts within a coded clinical event. These dimensions are System (CSY), Function (CFU), Etiology (CET), and Topography (CTO). The next section discusses the detailed makeup of each dimension. The Core dimensions place the coded clinical event in a specific medical space.

**Identifier Dimensions**

A group of Identifier dimensions places the coded clinical event in a specific patient space, i.e., they identify the patient and how...
<table>
<thead>
<tr>
<th>DIMENSION NAME</th>
<th>ID</th>
<th>MEANING IN PROBLEM-DEFINITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Dimensions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System</td>
<td>CSY</td>
<td>System involved in problem</td>
</tr>
<tr>
<td>Topography</td>
<td>CTO</td>
<td>Body region involved in problem</td>
</tr>
<tr>
<td>Function</td>
<td>CFU</td>
<td>Medical significance predicate</td>
</tr>
<tr>
<td>Etiology</td>
<td>CET</td>
<td>Cause of problem</td>
</tr>
<tr>
<td>Identifier Dimensions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Type</td>
<td>IET</td>
<td>Specific patient identification of problem</td>
</tr>
<tr>
<td>Descriptor Dimensions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activities</td>
<td>DAC</td>
<td>Descriptors of problem</td>
</tr>
<tr>
<td>Laterality</td>
<td>DLT</td>
<td>Activities related to problem</td>
</tr>
<tr>
<td>Location</td>
<td>DLO</td>
<td>Bodily direction of problem</td>
</tr>
<tr>
<td>Medications</td>
<td>DMD</td>
<td>Setting related to problem</td>
</tr>
<tr>
<td>Time Mode Onset</td>
<td>DMO</td>
<td>Therapeutic agents or drugs</td>
</tr>
<tr>
<td>Patient Perspective</td>
<td>DPT</td>
<td>Rate of problem onset</td>
</tr>
<tr>
<td>Quantity</td>
<td>DQT</td>
<td>How patient views problem</td>
</tr>
<tr>
<td>Qualifier</td>
<td>DQU</td>
<td>Quantity involved in problem</td>
</tr>
<tr>
<td>Status</td>
<td>DST</td>
<td>Meaning change indicator such as negation</td>
</tr>
<tr>
<td>Severity</td>
<td>DSV</td>
<td>State of problem</td>
</tr>
<tr>
<td>Time Base</td>
<td>DTB</td>
<td>Severity of problem</td>
</tr>
<tr>
<td>Time Duration</td>
<td>DTD</td>
<td>Activities related to problem</td>
</tr>
<tr>
<td>Time Onset</td>
<td>DTO</td>
<td>Duration of one problem episode</td>
</tr>
<tr>
<td>Time Periodicity</td>
<td>DTP</td>
<td>Time when problem occurs or occurred</td>
</tr>
<tr>
<td>Units</td>
<td>DUT</td>
<td>Frequency of episodes</td>
</tr>
<tr>
<td>Who</td>
<td>DWH</td>
<td>Reference to person other than patient</td>
</tr>
<tr>
<td>Modifier Dimensions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Color</td>
<td>MCC</td>
<td>Extra modifiers of a problem</td>
</tr>
<tr>
<td>Comment</td>
<td>MCT</td>
<td>Color modifier</td>
</tr>
<tr>
<td>Shape</td>
<td>MSH</td>
<td>Extra physician comment</td>
</tr>
<tr>
<td>Size</td>
<td>MSI</td>
<td>Shape modifier</td>
</tr>
<tr>
<td>Texture</td>
<td>MTX</td>
<td>Size modifier</td>
</tr>
<tr>
<td>Specifics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuation</td>
<td>SCO</td>
<td>Storage specifications</td>
</tr>
</tbody>
</table>

Figure 7-1. List of Anamnestic Coding Scheme Dimensions with a Brief Description of Their Meanings in Problem-definitions
he/she fits into the medical setting. None of the CHCP problem-definitions had patient identifying information. Therefore, only one Identifier dimension, Event Type (IET), is used. In an automated medical record system, there would be Identifier dimensions to identify a particular patient and specify the time of the encounter. For example, one Identifier dimension would be Patient Number.

**Descriptor Dimensions**

Descriptor dimensions set the coded clinical event in time and place and contain detailed clinical information. These dimensions are Activities (DAC), Laterality (DLT), Location (DLO), Medications (DMD), Time Mode of Onset (DMO), Patient Perspective (DPT), Quantity (DQT), Qualifier (DQU), Status (DST), Severity (DSV), Time Base (DTB), Time Duration (DTD), Time Onset (DTO), Time Periodicity (DTP), Units (DUT), and Who (WHO). A full description of these dimensions is given in the next section.

**Modifier Dimensions**

Modifier dimensions are used to code additional information which modifies the Function code or the entire coded clinical event, but is unnecessary for at least a general understanding of the event. There are five Modifier dimensions: Color (MCO), Comment (MCT), Shape (MSH), Size (MSI), and Texture (MTX).

**Specifics Dimension**

The last group of dimensions, Specifics, provides further details for storage and logical linkage between coded clinical events. There is only one dimension in this group--Continuation (SCO).

A total of 27 dimensions encode the CHCP problem-definitions. ACS
places coded clinical events on a much finer grid than any other medical coding scheme. SNOP uses four dimensions and SNOMED uses six for diagnoses. The large number of dimensions in ACS is indicative of the domain of information ACS orders and represents.

Description of Dimensions

System

The System dimension is a Core dimension corresponding to the Topography dimension in SNOP and SNOMED. It is based on the traditional Medical Systems Review which divides the human body into systems. An oversimplified explanation of a medical "system" is a unit whose components perform a task which contributes to a physiologic function. Medicine has found it convenient to organize the body into systems and many medical specialties deal primarily with one system, but no system is completely isolated from another. The Cardiovascular and Nervous system, for example, pervade every other system.

ACS has 15 major divisions within the System dimension: Psychosocial; Nervous; Eye; Ear; Respiratory; Cardiovascular; Digestive; Urogenital; Endocrine; Hematocrit, Reticuloendothelial; Skin; Musculoskeletal; Body Regions; and Whole Body. No single definitive rule exists for medically organizing the body into systems. Some rules-of-thumb are based on (1) physiologic function, (2) type of tissue, (3) medical specialty area, or (4) convenience. Even though no decisive algorithm can be found for dividing the human body into systems, these traditional systems suffice and give some modular perspective to a very complicated total system, the human body.
Topography

Topography is another Core dimension; it is a subset of the System dimension. Primarily, Topography is a copy of the Body Regions within the System dimension which locates a problem on the human body. Many problems involving skin take both a System code and a Topography code. Rather than having subsystems of Arm Skin, Elbow Skin, Neck Skin, Leg Skin, etc., under the Skin system, ACS uses the inherent symmetry to put Skin within the System dimension and Arm, Elbow, Neck, and Leg within the Topography dimension. ACS uses the Topography dimension whenever there is both a System and a Body Region specified.

Function

The Function dimension is an innovation in ACS and, as with the definition of System, all definitions are inadequate. There is no difficulty identifying a Function dimension code within a problem-definition, but each of three researchers actively involved with ACS produced three different definitions for the Function dimension per se. Moreover, each of the researchers took exception with the other’s definition and even recognized inadequacies with their own definitions. Therefore, in describing the Function dimension, some statements apply generally, but have exceptions.

The Function dimension has 14 large categories: Basic Function; Dysfunctions/Abnormalities; Pain; Inflammation and Infection; Fibrosis; Vascularity, Tissue Fluid; Mass/Tumor; Size and Number; Structure, Stability and Anatomical; Physical Characteristics; Diagnostic Procedure; Therapy; Surgical Procedure; and Administrative Procedure. The Basic Function category is generally used to indicate the direction of change
in a physiologic function which is coded in the System dimension. ACS was originally designed to code signs and symptoms and the Basic Function category is especially useful for coding these. For example, CONSTIPATION encodes into (CSY6500-"Defecation," CFU1400-"Decreased Function"). ACS takes advantage of the repetition inherent in many diverse medical problems since the problem is often based upon an increase, decrease, or cessation of a particular physiologic function.

Ideally, the Function dimension contains primitive semantic units representative of very basic dysfunctions. In other words, it is exceptionally adept at coding single manifestations. However, not all problem definitions are simple manifestations such as signs and symptoms. Some are diagnoses which are labels for sets of signs and symptoms. Wherever possible, a diagnosis is coded according to its simple definition. For example, HEPATOMEGALLY, codes to (CSY6C20-"Liver," CFU8600-"Size Increase").

When there is no simple definition, ACS uses the major manifestation. For example, ANEMIA codes to (CSY9620-"Hemoglobin," CFU1400-"Decreased Function"). The objective is to make each coded problem unique while maintaining the ability to retrieve, with a minimum number of keys, all problems pertaining to a general topic.

In some instances, one dysfunction occurs with only one system. COUGH is an example of such a dysfunction. A cough involves the respiratory system and it is a reflex movement; but a hiccup also fits this description. Therefore, to maintain uniqueness in the coded representation, the Function dimension contains separate codes for hiccup and cough.
In still other instances, there are diagnoses which have no do­minant manifestations, but are labels for a well-integrated set of signs and symptoms. Names of syndromes are an example of these types of diagnoses. Medicine uses, for example, "dumping syndrome" and "organic brain syndrome," to stand for a set of observed signs and symptoms. As more knowledge is gained about these syndromes, they are often relabeled with more appropriate or descriptive names. In encoding these diagnoses into ACS, the System dimension code is often the major system involved, or it is simply coded as CSYD800-"Systemic Disorders." The Function dimension code is more than a semantic primitive in such cases; it often is the full name of the diagnosis placed in the hierarchy close to some similar dysfunction.

The Function dimension may be thought of as the predicate in a coded clinical event because it describes the condition, or state of the patient. The Function dimension also contains Diagnostic, Therapeutic, and Surgical Procedures. Rarely are these legitimate in statements of medical problems; however, they are vital for coding other types of clinical events into ACS and they allow for encoding CHCP problem-definitions which are, in reality, history statements or procedures rather than problems.

Etiology

The last Core dimension is Etiology. It corresponds to the Etiology dimension of SNOP and SNOMED. The dimensions discussed thus far, System, Topography, and Function, all had received extensive development in prior work by Brunjes (1971), Lynch (1975), and Powsner (1978). With these dimensions, this research merely added details. The Etiology
dimension was the first dimension which had no initial framework.

SNOP and SNOMED's Etiology dimension were too detailed for present purposes. Some of these details may be added later for ACS to code a complete medical record; but many of the microscopic Etiologies of SNOP and some of the broad categories were either too detailed or too general for use with ACS in coding the CHCP problem-definitions. Therefore, the current Etiology dimension was developed from a composite of concepts in SNOP and SNOMED, and the list of CHCP problem-definitions. The Infective Organisms category was taken almost entirely from *Practical Concepts in Human Disease* by Bickley (1974). The organization of other categories, such as Environmental Agents, Chemicals and Chemical Compounds, Plants, Food and Food Products, Allergy/Atopy, Etiology Associated with Other Dimensions, Accident, Congenital, and Aging were all dictated by etiologies, or causes, found in the CHCP problem-definitions.

Most of the Etiology categories are self-explanatory; however, Etiology Associated with Other Dimensions requires further explanation. The best explanation is by example. MASTECTOMY PAIN is a problem-definition where the pain is the result of an operation. Here the etiology is a complete coded clinical event, and it is coded using CETA200-"Etiology is System and Function" and the Continuation dimension code, SC01000-"Etiology Continuation."

The Etiology dimension is not complete. It, as well as the other dimensions originating in this work, need much more extensive use before they are fully developed. However, the present research does provide a framework upon which to build.
Event Type

For purposes of encoding the CHCP problem-definitions, there is only one Identifier dimension, Event Type. Event Type designates the medical setting which the coded clinical event describes. As such, it indicates the logical relations among dimension codes. These are discussed in the last section of this chapter.

One may initially assume all event types in the CHCP set of problem-definitions are IET8000-"Problem." However, the content of problem-definitions is in practice far from what Lawrence Weed (1976) defines as a medical problem. Some problem-definitions are procedures and several are history statements. To signify that these diverse statements are made within the problem list, many event types are subordinated under the Event Type, IET8000-"Problem." In an automated medical record system using ACS, when these event types are input from different sources such as a history form or doctor's orders, a separate code for the event type should be used. However, the present arrangement of the Event Type dimension is appropriate for coding problem-definitions.

Activities

The first Descriptor dimension to be discussed is Activities. Several CHCP problem-definitions contain reference to common activities in which people engage. For example, IMMUNIZATION FOR TRAVEL and HEADACHES AT WORK. In the first example, travel is really an etiology or cause of the immunization. The ACS code contains CETA400-"Etiology is for Activity" and DAC3E30-"Travel." In the second example, the code contains DTO7300-"Indefinite Time around Base," DTB9000-"Activity," and DAC3730-"Work." Time Onset, DTO, and Time Base, DTB, will be explained.
later. Information coded in Activities is seldom necessary for understanding the general nature of a problem, which is why Activities is grouped in the Descriptor dimensions. However, the Activities dimension does enable mapping some detailed information about the patient into ACS.

**Laterality**

Laterality is a Descriptor dimension specifying direction in reference to System or Topography. The codes are arranged so that a logical OR of the bits from more than one Laterality code within a coded clinical event produces the desired code. For example, DLT1000—"Right" OR'ed with DLT2000—"Left" results in DLT3000—"Bilateral." The code for "Lower, Left," DLT2200 is an OR of DLT2000—"Left" and DLT0200—"Lower." If there are multiple directions given in problem-definitions, the arrangement of the Laterality dimension allows for a quick collapsing into one Laterality code.

**Location**

Location is a dimension within the Descriptor category which is rarely used; but it does allow for encoding information about places which people normally frequent. One example of its use is in the problem-definition, SCHOOL AVOIDANCE, which codes as (CSY0450—"Avoidance," CFU2220—"Abnormality/Disorder," DLO5300—"School").

**Medication**

The Medication dimension is one Descriptor dimension which is totally undeveloped in this research. Arranging medications into a hierarchy is a large project. The current automatic encoder merely identifies the few medications found in the CHCP problem-definitions. A much more extensive list of medications could be found in event types, prescriptions
or doctor's orders. It must be noted, however, that physical as well as chemical medications are coded into the Medication dimension. In addition, the Medication dimension contains nutritional substances and supplements such as vitamins and minerals.

**Time Mode Onset**

Time Mode of Onset is one of five dimensions dealing with time in the group of Descriptor dimensions. Presently, it contains only two codes, Sudden and Gradual. In medicine, the difference in the rate of onset of a problem is often important. Time Mode of Onset captures this information.

**Patient Perspective**

Patient Perspective is a Descriptor dimension which stores information descriptive of the problem from the patient's point of view. A very few of the CHCP problem-definitions need this dimension to specify the patient's perspective of the problem. Patient Perspective is used for encoding problem-definitions such as DESIRES CIRCUMCISION, THINKS SHE AS HEPATITIS, and FEELS NERVOUS.

**Quantity**

The Descriptor dimension, Quantity, stores definite and indefinite numbers. A definite number is represented by a set of digits and may be any number, such as a test result or quantity of pills. Indefinite quantities are on an ordinal scale from DQT0110-"A Fraction Implied" to DQT0170-"Many Implied."

**Qualifier**

The Qualifier dimension is a Descriptor dimension storing any information which changes the intent of a problem-definition. One way to
change the intent of a problem-definition is to add a comment, such as, LIKELY, RESEMBLES, QUESTION OF, PROBABLE, or SUSPECTED. These comments change the problem-definition from a positive statement of fact to a qualified statement. Negation of a clinical event is also indicated in the Qualifier dimension.

The Qualifier dimension also stores a rather indefinite term, STATUS POST. Many CHOP problem-definitions use this term which seems to mean a state arrived at after the occurrence of some event. For example, STATUS POST is often associated with operations, such as STATUS POST MASTECTOMY, STATUS POST POLYPECTOMY, and STATUS POST HERNIÆRRAPHY. Encoding as post operative checkups is not correct since there is also some use without operations, such as STATUS POST MYOCARDIAL INFARCTION, STATUS POST ULCER, and STATUS POST DISLOCATION R WRIST. These shed some doubt as to whether the prefix STATUS POST, means a checkup. STATUS POST codes into the Qualifier dimension because it definitely changes the meaning of the rest of the problem-definition from a description of an occurrence to the statement of having arrived at some state after the occurrence. Without some standard medical usage of the term, this seems to be the most appropriate dimension to store STATUS POST.

**Status**

The CHOP format for problem-definitions places most of the Status dimension concepts within a comment occurring at the end of a line of text and often separated from the text by a semicolon. These concepts are remarks about the state of a problem in general. For example, the problem may be BETTER, RESOLVED, or WORSE or it may demonstrate NO CHANGE, be SYMPTOMATIC, or ASYMPTOMATIC. The Status dimension stores
these concepts as well as several concepts which are dependent upon both
the institution where they are used and the problem which they modify.

"Types," "degrees," or "stages" are highly institution dependent. For example, a research hospital may have a well-defined set of stages for cancer which relate to treatment and prognosis. Another general hospital may not "stage" cancer at all or may "stage" it using different meanings. Therefore, these state descriptors are very institution dependent. They are also dependent upon the particular problems with which they are used. For example, a staging of hypertension does not have the same implications as a staging of malignant melanoma. Since the occurrence of these state descriptors in the CHCP problem-definitions were too few to structure, they are all lumped into one Status code, DSTF000-
"Status to be classified later."

Severity

Severity was a well-defined dimension in the CHCP signs and symptoms. There were four Severity codes: Absent, Mild, Moderate, and Severe, printed on the CVF form for use with signs and symptoms. Five additional Severity codes were needed to code the CHCP problem-definitions. They are Positive, Negative, Abnormal, Benign, and Normal. These represent general concepts which apply to many problems. Of course, they do overlap the domain of Status; however, Status is descriptive of an entire problem, whereas Severity describes one manifestation.

Time Base

The Descriptor dimension, Time Base, denotes a reference point in time. It contains the Calendar Cycle, the Life Cycle, Weather Cycle, Feeding Cycle, Sleeping Cycle, and reference to other dimensions.
Medical problems do not always occur in reference to a specific date and hour. Problems may be of the nature of SEASONAL RHINITIS, NOCTURNAL HEADACHES, or POST MENOPAUSAL BLEEDING. These problems specify a time, but it is not conveniently stored in any form based on computer wall-clock time. A fully automated medical record system may look-up the approximate time of menopause, for example, in its store of patient information, and assign at least an approximate date to the bleeding problem. However, no such computation can be made with SEASONAL and NOCTURNAL. Therefore, the dimension, Time Base, provides for storage of time information which does not reference a specific date or hour.

**Time Duration**

ACS uses the dimension, Time Duration, to code the length of a clinical event episode. The length may be a definite number or some indefinite measure, such as DTD7300-"Brief Duration Implied" or DTD7700-"From Onset to Present (Constant)."

**Time Onset**

Time Onset is used to specify when a problem began or was first noticed. It, too, may be a definite time or an indefinite time, such as DT07300-"Indefinite Time around Base" or DT07700-"Indefinite Time before Base." The base refers to the dimension, Time Base.

**Time Periodicity**

Time Periodicity is the last time related dimension in the set of Descriptor dimensions. It conveys either a definite number of occurrences of a clinical event or the periodicity with which the event occurs. Examples of indefinite time within Time Periodicity are DTP7300-"Irregular," DTP7353-"Few Occurrences Implied," and DTP7700-"Regular (Periodic)."
Units

The Descriptor dimension, Units, presently contains only time units. In the design set of CHCP problem definitions, there were no other units, such as length. For general coding, the Units dimension must be expanded to store many different units of measure.

Who

Occasionally, a problem definition refers to a person other than the patient. For example, FAMILY HISTORY OF DIABETES. The Descriptor dimension, Who, enables the coding of references to persons other than the patient. Without the presence of the Who dimension, all remarks made in a problem definition are assumed to concern only the patient under treatment.

Color, Comment, Shape, Size, and Texture

The five Modifier dimensions need very little explanation since their names are very indicative of their meaning. The Comment dimension presently has only two codes, MCT1000-"Confidential," and MCT2000-"See Dictation." The other Modifier dimensions use codes which are actually subsets of the Function dimension. Modifier dimensions are coded with Function in the same way that Topography is coded with System, i.e., if the word dictionary produces two Function codes for one coded clinical event, and one of these falls within the range of the Function codes which may be Modifiers, then the latter Function code is changed to a Modifier dimension.

Examples provide further clarification of the Modifier dimensions. The problem definition, MOSAIC WART, uses WART for the Function code and MOSAIC is changed to the Color code, MC0A6A0. NUMMULAR DERMATITIS
produces the Shape code, MSH94F3. PILONIDAL CYST codes to the Texture code, MTXA3B3, and SMALL MASS uses Size, MSI8A30. When, for example, the word "small" occurs in SMALL STATURE, the Function code is CFU8A30, and no conversion to the Modifier dimension, Size, takes place. In other Modifier dimensions, except Comment, no transformation occurs without the presence of another Function dimension.

Continuation

The Specifics category of dimensions contains one dimension, Continuation, which specifies further details for storage of the coded clinical events. The Continuation dimension is used whenever it is necessary to have more than one code for any dimension within a coded clinical event. There are presently only three types of situations where it is necessary to use the Continuation dimension.

The first Continuation code, SC00100, is used whenever concepts within a coded clinical event require either two System or two Function codes. The problem-definition, COLOVAGINAL FISTULA, needs two System codes, one for COLON, and one for VAGINA. The encoder produces two coded clinical events for this problem-definition. One has the Continuation code, SC00100.

When two Function codes are needed, SC00100 is also used. Generally, one of the Function codes is CFU4200-"Inflammation" or CFU4400-"Infection." For example, coding INFECTED CYST, requires a Function code for Infection and Cyst. To maintain the requirement of only one code per dimension in a coded clinical event, the encoder produces two coded clinical events, one of which has linkage indicated by the SC00100 Continuation code.
The second Continuation code, SC01000, is used whenever more than one coded clinical event is required to represent etiology. This may occur in several ways. In DIABETES NEPHROPATHY, diabetes is the etiology, or cause, of the kidney disease. The main coded clinical event codes kidney disease with an Etiology code of CETA200-"Etiology is from System and Function." The secondary coded clinical event contains SC01000-"Etiology Continuation and the System and Function" with the code for diabetes. TRAUMA CAR ACCIDENT codes as (CFU9000-"Trauma," CETA300-"Etiology is Function") and (SC01000-"Etiology Continuation," CET4330-"Automobile"). By linking, a very long list of causes may be coded.

The last Continuation link is used when Time dimensions need to incorporate additional dimensions already existent in the coded clinical event. For example, CANCER RECTUM AFTER COLOSTOMY encodes into the two coded clinical events: (CSY6480-"Rectum," CFU7630-"Neoplasm, NOS, Malignant," DT07500-"Indefinite Time After Base," DTBB000-"Base is System and Function") and (SC02000-"When Occurs Continuation," IET8300-"Surgical Procedure," CSY75B0-"Large Bowel," CFUE6D0-"Stoma Creation").

Relations Among Dimensions

The previous section described each dimension. This section will discuss how the dimensional codes may be used to represent all the explicit information in Natural Language statements of medical problems. In addition, a sketch of how ACS is used with other types of clinical events will be given to provide the reader with a perspective of an automated medical record system using ACS.

Figure 7-2 displays sample problem-definitions from the design
ACTIVE DUODENTAL ULCER
DSTP000-"Status to be classified later"
CSY6380-"Duodenum"
CFU42CO-"Ulceration"

ACUTE BRONCHITIS
DM03000-"Sudden onset"
CSY4620-"Bronchial passages"
CFU4200-"Inflammation"

ALLERGIC TO ASPIRIN
IET8500-"History Statement"
CFU2F40-"Drug reaction"
DMD0000-"Medication to be Classified Later"

COUGH SEVERAL DAYS
CSY4244-"Expiratory movts/exhalation"
CFU2C12-"Cough"
DTO3500-"Definite time before base"
DTB3000-"Current visit"
DTD7700-"From onset to present (constant)"
DQT0150-"Moderate quantity implied"
DUT5900-"Day"

INFECTED BURN R FOOT
CSYA000-"Skin"
CFU4400-"Infection"
CTOC5CO-"Foot, region"
DLT1000-"Right"
CSTA300-"Etiology is function"
AND
SCO1000-"Etiology continuation"
CFU9C20-"Burn"

H/0 RECURRENT UTI
IET8500-"History statement"
CSY7200-"Urological"
CFU4400-"Infection"
DTP7355-"Many occurrences implied"

Figure 7-2. Sample Problem-Definitions from the Design Set With Their Interpreted Anamnestic Coding Scheme Representation
MILD CHEST PAIN
DSV1000-"Mild"
CSYC800-"Chest"
CFU3000-"Pain"

MOTHER HAS CANCER
DWH1335-"Mother"
DTO7300-"Indefinite time around base"
DTB3000-"Current visit"
CFU7680-"Neoplasm, NOS, malignant"

SEIZURES HOME
CSY1340-"Brain"
CFU2610-"Seizure"
DLO3300-"Home"

SMALL MASS
MSI8A30-"Small"
CFU7000-"Mass/tumor"

SOFT CYST LIKE GROWTH
MTXA340-"Soft"
CFU7200-"Cyst"
DQU0390-"Resembles or symptomatic of"

STRESS INCONTINUENCE
CSY7244-"Bladder sphincter"
CFU2C00-"Altered movements or control"
DTO7300-"Indefinite time around base"
DTB9000-"Activity"
DAC6A00-"Stress"

WANTS COSMETIC SURGERY ON NOSE
IET8300-"Surgical procedure"
CSY4430-"Nose"
CFUE300-"Cosmetic Surgery"
DPT3000-"Desires"

Figure 7-2 (con't). Sample Problem-Definitions from the Design Set With Their Interpreted Anamnestic Coding Scheme Representation
set with the interpreted codes produced by the automatic encoder. By referring to the list of dimensions in Figure 7-1 and the previous discussion, the reader should be able to evaluate for himself/herself whether the dimensional codes capture all the main concepts explicitly evident in the problem-definitions.

The codes alone do not reflect all the explicit information in these problem-definitions because logical relations among codes are needed to completely specify the problem-definition. Figure 7-3 shows some of the logical relations among dimensions and/or groups of dimensions for coding problem-definitions. Each rectangle is labeled with the name of the group of dimensions from which the dimensions within the box are taken. Arrows are labeled with an indication of the logical relations. Arrows which terminate within a rectangle indicate relations involving the specific dimensions near their termination. Arrows which terminate outside a rectangle denote relations involving all dimensions in the box. In the succeeding discussion the dimensions within rectangles will be referenced by the lower case letter in the upper right corner.

The Core dimensions, box d, contain the central concepts of the problem-definition. Each dimension in this group is related to another dimension within the group. Etiology is the causative agent for the Function; Function predicates System; and Topography specifies a location for System. All other dimensions are either related to particular dimensions or to the Core dimensions as a group.

The Identifier dimensions, box a, are related to the entire coded clinical event so they are shown with an arrow pointing toward the group of Core dimensions. Recall that only one Identifier dimension, Event Type,
Figure 7-3. Diagram of Logical Relations Among Some Dimensions
is present in the design set. Other Identifier dimensions such as Patient Number were unnecessary for the current investigation.

The Modifier dimensions, boxes b and c, contain information which is unnecessary for general understanding of the coded clinical event. Color, Shape, Size, and Texture modify Function; whereas Comment modifies the entire coded clinical event.

The Descriptor dimensions, Activities and Medication, in box e, are related to Etiology. They actually supply a name to an activity or medication which acts as a causative agent for the Function. Activities is a separate dimension because Time Base as well as Etiology references it. Medication is a separate dimension because it has a different role in other clinical event types. For example, concepts within Medication are agents in some therapeutic procedures under the Event Type, doctor's orders. In addition, certain retrievals might need to isolate all coded clinical events involving concepts in these dimensions. For example, one request might be for all coded clinical events related to the Medication concept, antibiotics, or the Activities concept, sports.

The Descriptor dimensions in box f are related to the entire coded clinical event. They specify detailed information in the coded problem-definition. One dimension in this box, Qualifier, is of particular importance in aggregate retrieval. Qualifier concepts indicate that the problem means something other than that given in the Core dimensions alone. Any aggregate retrieval must check the Qualifier dimension before including the coded clinical event in output.

At the onset of this investigation, the Qualifier dimension was intended to be the only dimension which an aggregate retrieval need inspect.
However, the present ACS coding scheme requires checking Who, Status, and Patient Perspective as well as Qualifier. All of these can indicate that the coded clinical event is not a direct, positive statement of a problem. Future investigations which evaluate retrieval mechanisms may need to have duplicates of the concepts in Who, Status, and Patient Perspective reflected in the Qualifier dimension.

Only the relation of the Descriptor dimensions involving time, box g, to the entire coded clinical event are shown in Figure 7-3. Figure 7-4 shows additional relations among time dimensions. The three time dimensions, Time Duration, Time Onset, and Time Periodicity, may contain either definite or indefinite times. Definite times are on interval scales. Indefinite times are on ordinal or nominal scales. Both definite and indefinite times may reference Time Base. Definite times may also use a definite Quantity and Units. Indefinite times may use an indefinite Quantity and Units. For example, 6 WK POST PARTUM CHECK needs a definite Quantity-"Integer, 6," Units-"Week," and Time Base-"Delivery," whereas HYPERTENSION MILD MANY YEARS needs an indefinite Quantity-"Many Implied" and Units-"Years."

Time Base may specify a concept in the Activities dimension or it may link to another coded clinical event just as Etiology may contain a link to another coded clinical event. The curving arrows toward these dimensions represent linking in both Figures 7-3 and 7-4. The additional curving arrow pointing toward the Core dimensions indicates that a con-joining link may also be made between two coded clinical events. This situation occurs for encoding problem-definitions such as VOMITED BLOOD which need two Function codes to represent all explicit information, i.e.,
Figure 7-4. Diagram of Logical Relations Among Dimensions Containing Time Concepts
vomiting and bleeding. The conjoining link is a compromise between having one Function code per coded clinical event and still encoding all information.

The last box in Figure 7-3, box h, contains the dimensions Quantity and Laterality which are related to System and Topography. Recall that Topography is only coded when the clinical event has a System code. In the presence of a Topography code, Quantity and Laterality are related to Topography; otherwise, they relate to System.

Some dimensions used in Figure 7-3 and 7-4 appear more than once, e.g., Quantity, Activities, Medication, Time Base, and Units. A link through Time Base or Etiology disambiguates the usage of Activities and Medication. The other dimensions are possibly ambiguous in a coded clinical event, i.e., they could appear more than once in a coded clinical event and their relations to the other dimensions within the coded clinical event could be unspecified. The CHCP problem-definitions did not require duplicates of these dimensions, but their relationship to other dimensions and the entire coded clinical event is unclear in some instances. Future work needs to have a separate Quantity dimension related to System and Topography, Time Duration, Time Onset, and Time Periodicity. Current research did not make these distinct dimensions because the concepts within the Quantity dimension were seldom used in the design set problem-definitions. Some similar provisions are needed to disambiguate Time Base and Units.

Figure 7-5 shows possible logical relations among dimensions for an Event Type other than problem-definition. This figure displays how some future investigation may use ACS to encode clinical laboratory test
Figure 7-5. Diagram of Proposed Logical Relations Among Dimensions for a Coded Clinical Event of the Type Laboratory Test
orders. The Function dimension contains the name of the test. A large set of data would be needed to confirm the usefulness of these relations; but, this figure is an example of how ACS may code other Event Types in the medical record. Notice that some of the same dimensions used to code problem-definitions also code laboratory tests.

This chapter has provided the reader with a detailed discussion of how ACS has been extended to encode the clinical event type problem-definition and how other clinical event types can be encoded into ACS for a complete ACS coded medical record. The next chapter discusses the design and implementation of the automatic encoder which transforms medical problems, stated in Natural Language, into ACS.
CHAPTER VIII

DESCRIPTION OF AUTOMATIC ENCODER

General Control Structure

The automatic encoder operates in two modes, manual and automatic. Manual mode allows the user to interact with the automatic encoder; automatic mode accepts no user direction after initial program options are set. Automatic mode provides the ability to automatically encode many problem-definitions, compare the automatically assigned codes with manually assigned codes, and tabulate results. In an actual medical setting, the mode of operation would be manual, with some modifications of the present program for a novice user.

Figure 8-1 is a flowchart of the main portion of the automatic encoder. The encoder continues to process when there are unfamiliar words in a problem-definition. This feature allows the user to view any partial results. In automatic mode, problem-definitions are written onto two different files depending upon whether the automatic encoding has been successful. Success is defined as having an automatically assigned set of codes matching a set of manually assigned codes and having all words defined in the problem-definition. As codes were manually assigned, they were appended to the end of each problem-definition. This made them available for comparisons with the automatically assigned codes when the encoder ran in automatic mode. In manual mode, the user has many options. The control characters and an explanation of their subsequent
While there are problem-definitions to read

Get the problem-definition to be coded

Preprocess?

Undefined words 

No undefined words

Set undefined word's flag

Try to assign codes

Were codes assigned and is there a set of manual codes?

Yes

Compare manual and automatically assigned codes

No

Automatic mode

Agreement with manual and no undefined words?

Yes

No

Display results and interact with user

p,m

q

Copy all remaining to failure file

Write to success file

Write to failure file

Write with manual

Tabulate results

Summarize

Figure 8-1. Flow of Control in the Automatic Encoder
COMMAND

return
p
m <code set>
s
t
q

INTERPRETATION

Tabulate results of the current problem-definition and proceed to input the next

Accept the automatically assigned code as the correct manual code, tabulate results, and proceed to input the next problem-definition

Put <code set> as the correct manual code

Re-display the original problem-definition and the interpreted codes

Try automatically encoding the current problem-definition again

Tabulate results, flush any remaining problem-definitions in the input file onto the output file and write the summary

DICTIONARY COMMANDS

du <word>
di <word>
eu <word>
ei <word>
kill <word>
ea <word>

RULE FILE COMMANDS

Display the uninterpreted definition of <word>
Display the interpreted definition of <word>
Edit the uninterpreted definition of <word>
Edit the interpreted definition of <word>
Remove <word> from the word dictionary
Add <word> to the word dictionary

Display rule i
Display all rules applied to the current problem-definition
Remove rule j from the set of rules
Move rule i to rule j
Edit rule j. If j is negative, then rule j is created.

Figure 8-2. Options Available in Manual Mode
actions are listed in Figure 8-2.

The automatic encoder, in manual mode, was used to assign codes to each of the 5,465 problem-definitions in the design set. Since the process of manually assigning codes using the automatic encoder may be a source of confusion, some further explanation of this process may be beneficial. A majority of the words in the design set had definitions in the word dictionary as a result of Powsner's (1978) work as well as this researcher's individual efforts with the word frequency list. Manual assignment of ACS codes means that the automatic encoder was used to review the codes assigned to the 5,465 unique problem-definitions in the design set.

The versatility of the automatic encoder design allows the user to either accept the automatically assigned code as correct or manually input a set of manually correct codes. The process of manually assigning codes is slow and tedious but the automatic encoder provides many conveniences. Each code is interpreted, thus there is no need to refer to a code book. In addition, the user has the options of editing the rules file or word dictionary and running the encoder again using the same problem-definition. Many of the problem-definitions were simple enough to create no difficulty. The medical consultant, Dr. Shannon Brunjes, provided clarification when questions arose concerning medical interpretation.

Figure 8-3 diagrams the preprocessing every problem-definition receives when it enters the encoder. The same subroutines that reduce the number of unique problem-definitions in the CHCP set preprocess each problem-definition. This maintains a link back to the CHCP
Normalize abbreviations

Strip standard comments and save their codes for later use

Define a word as any string of non-blanks separated by blanks

<table>
<thead>
<tr>
<th>Lookup any undefined words in word dictionary?</th>
<th>No words undefined</th>
</tr>
</thead>
<tbody>
<tr>
<td>At least one word undefined</td>
<td></td>
</tr>
<tr>
<td>First lookup iteration?</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Do some words have numbers?</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Try to identify the numbers and assign a code</td>
<td>Ø</td>
</tr>
<tr>
<td>If there is punctuation, identify it and separate it from any words</td>
<td>Ø</td>
</tr>
</tbody>
</table>

Return indicating no undefined words

For four iterations

Return indicating there are undefined words

Figure 8-3. Flow of Control in the Preprocessor
problem-definitions in their original form.

The subroutine used to look up every word in the word dictionary employs the hashing method designed by Powsner (1978). Initially, words are defined as strings of characters delimited by blanks. A maximum of four iterations attempt to locate each word in the word dictionary. On the first iteration, the preprocessor tests for digits in undefined words. If there are any digits, it calls a number handling subroutine which attempts to identify each word with at least one digit.

The number handling subroutine tries to find a pattern which matches any word containing a digit. For example, the pattern "dd/dd/dd," where "d" stands for a digit, is tentatively determined to be a date. If a date test subroutine confirms that it is a legitimate date, then the encoder assigns a code for a date. The automatic encoder, in its present form, only identifies dates. The form for date storage, as well as for storage of other numbers, is so dependent upon a particular implementation of an automated medical record system that no decision was reached for final storage procedures.

Many of the CHCP problem-definitions had nonstandard spacing with punctuation. For example, NUMBNESS R ARM-LEG, ?UTI, and C/SECTION-EMERGENCY. In the first iteration, each word is any string of characters separated by blanks. Therefore, ?UTI is searched for as one word in the word dictionary. Any words not found in the word dictionary are subjected to a test for punctuation. When punctuation is encountered, it is used as a word delimiter and each word is again sought in the word dictionary. For example, ?UTI becomes two words, ? and UTI. Inconsistencies in spacing and punctuation within CHCP problem-definitions are most likely
the result of both keypuncher interpretation and discretion.

Figure 8-4 demonstrates the flow of control used in actually assigning a set of codes to a problem-definition. Of necessity, this flowchart is greatly simplified. The automatic encoder's first objective is to satisfy any tests found in the word dictionary. The next section of this chapter fully describes the format of the word dictionary. In effect, the automatic encoder tries all combinations of word sense meanings in the word dictionary for each word in the problem-definition. Some numbers and punctuation are considered words at this point, but their corresponding ACS codes are fixed.

When a set of ACS codes completely satisfies the word dictionary tests, the automatic encoder proceeds to another set of subroutines, the rules section. Some rules are stored on disk and their grammar is described in the following section of this chapter. Other rules are hard coded into subroutines to increase their speed of computation.

There are three hard coded rule subroutines. The first OR's all Laterality dimensions in a coded clinical event. The purpose of this was discussed in Chapter VII; it changes DLT1000-"Left" and DLT2000-"Right" to DLT3000-"Bilateral." The second rule subroutine eliminates multiple codes from one dimension which have a hierarchically subordinate code already present within the coded clinical event. The last rule subroutine eliminates duplicate codes from any coded clinical event.

Presently, there are only two reasons for rejecting a set of ACS codes in the rules section. A rejection may occur if the coded clinical event does not have a Function code and has either no Continuation dimension code or a Continuation code of SC00100-"Continuation of Conjoined
Figure 8-4. Flow of Control Used to Automatically Assign Codes
The only other rejection in the rules section occurs when a coded clinical event has more than one code for any dimension. If the rules section rejects a set of dimension codes, the automatic encoder tries to find another set of codes acceptable to the word dictionary.

The automatic encoder has the ability to request assistance from the user in the rules section; however, this feature was not used in the design set. When there is a reason for rejection in the rules section, the computer simply outputs an error message. The whole question of the level of interaction to provide a novice user is beyond the scope of this dissertation. Ideally, there should be an extensive set of entries and exits, with the automatic encoder requesting information from the user. Current evidence indicates implementation is easily achieved; however, the content of such a conversation is a difficult question to answer.

An important consideration in conversations with a user is where to place responsibility. An automatic encoder could merely accept input and either encode it or refuse to encode it. With a few exceptions, this describes the functions of the presently designed automatic encoder. On another level, the automatic encoder could be responsible for completeness of information. For example, any problem-definition containing CYST should have some information locating the cyst on the human body. An "intelligent" automatic encoder would ask the user for a location. Of course, considering legal requirements this almost requires the user to be the attending physician. Since medicine is not currently practiced with physicians using computer terminals, the automatic encoder described here mimics a transcriptionist more than it does a physician's aid.
Description of Word Dictionary

The word dictionary contains ACS codes for each word sense meaning. The purpose of the word dictionary is to use information from neighboring words to select a word sense meaning for the word under consideration; and to give instructions to the encoder based on the word sense meaning. An entry for each word in the word dictionary is composed of a (test, instruction) pair, which will be called a definition. A definition has the following form:

\[
\text{<definition>} ::= \text{<test>;<instruction>}
\]

\[
[;\text{<test>;<instruction>}]\]

where brackets, [], mean optional repetition; ::= is a definitional metasymbol; a lowercase letter, b, is a literal space; angled brackets, <>, denote a type; and all other symbols are literals.

A <test> is composed of conjoined or disjoined <telement>s. The form of a <test> is:

\[
\text{<test>} ::= \text{<telement>}[|\text{<telement>}]\wedge
\]

\[
\text{<telement>}[&\text{<telement>}]\]

where \(\wedge\) is a metasymbol for OR. Note that conjunction, \&, and disjunction, |, cannot occur within the same <test>.

A <telement> has the form:
Note that a <telement> may be negated as denoted by !.

The meanings of the prefixes, O_, F_, N_, M_, and A_ are as follows:

1. O_ means if the <templet> occurs anywhere within the problem-definition, then execute the corresponding <instruction>.
2. F_ means if the <templet>s occur in the specified order around the defined word, denoted by "X," then execute the corresponding <instruction>.
3. N_ is a no-op test; in all cases the corresponding instruction will be executed.
4. M_ is not a real test; it means that the word being defined is a misspelling of another word given in <word>. The encoder will automatically look-up <word> and use its definition for the misspelled word.
5. A_ is similar to M_, but it means that the word being defined is an alternate form of <word>. A_ performs the function of a suffix processor. For
example, ABDOMINAL, is an alternate form of ABDOMEN; therefore, A_ABDOMEN is a <telement> for ABDOMINAL, and instructs the dictionary look-up routine to use the ABDOMEN definition.

A <templet> has the following form:

\[
\text{<templet> ::= W_ <word> ^T_ <type> ^} \\
\text{C_ <code>
}\]

A <word> is a string of characters which is a literal word. A <type> is a number designating a type of word, e.g., punctuation or number. Figure 8-5 gives the different numbers in <type> and their meaning.

A <code> is a seven character dimension code. The first character is a letter designating the dimension class Identifier (I), Core (C), Descriptor (D), Modifier (M), Specifics (S). The next two characters are letters designating the specific dimensions within each class. Chapter VII provided a thorough description of each dimension and Figure 7-1 showed the pneumonics for each dimension. The next four characters are hexadecimal digits. They are the actual numeric code within a dimension. Within <code>, any hexadecimal digit which is replaced by "Z" matches any hexadecimal digit.

The preceding defines the test portion of a definition; the instruction part uses some similar constructs. An <instruction> has the form:
<table>
<thead>
<tr>
<th>NUMBER TYPE</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>Date</td>
</tr>
<tr>
<td>103</td>
<td>Possible date</td>
</tr>
<tr>
<td>104</td>
<td>Integer</td>
</tr>
<tr>
<td>105</td>
<td>Social security number</td>
</tr>
<tr>
<td>106</td>
<td>Real</td>
</tr>
<tr>
<td>107</td>
<td>Severity (1+, 2+, etc.)</td>
</tr>
<tr>
<td>108</td>
<td>Problem number (#1)</td>
</tr>
<tr>
<td>109</td>
<td>Blood pressure (120/90)</td>
</tr>
<tr>
<td>110</td>
<td>Vision (20/200)</td>
</tr>
<tr>
<td>112</td>
<td>Unidentified number</td>
</tr>
<tr>
<td>113</td>
<td>Size (3x2)</td>
</tr>
<tr>
<td>114</td>
<td>Roman Numeral</td>
</tr>
<tr>
<td>115</td>
<td>Year Only</td>
</tr>
<tr>
<td>116</td>
<td>Number with comma (&quot;4,&quot; may be used in May 4, 1978)</td>
</tr>
</tbody>
</table>

Figure 8-5. Types of Numbers Recognized by the Automatic Encoder and Their Meaning
<instruction> ::= (E_<code>
    [({,+}*<code>}]^  
    R_<word>^A_<literal>^  
    P_T_<type>) [,<instruction>]

Each component of <instruction> has the following explanation.

(1) E_ means assign the following codes to the word being defined.

(2) R_ means remove <word> from further consideration. This feature allows the removal of any words from the idiomatic expressions.

(3) A_ means print the <literal> for the user to see. Generally, <literal>, is a question seeking additional information from the user. In a working encoder, this feature would be much more important than it is in the present implementation.

(4) P_T_ means assign to the word the type label given in <type>.

This concludes formal discussion of dictionary structure. To avoid complete confusion, sample dictionary entries in Figure 8-6 provide clarification. The dictionary entries shown in Figure 8-6 differ in format from the actual way the definition appears on disk. Specifically, extra line feeds provide a readable display and each dimension code has its interpretation in double quotes beside it.
COLD
F_X W_SORE |
F_X W_SORES;
  R_SORE.R_SORES.1_CSX6214 "ORAL COMMISSURE",
  CFU4400 "INFECTION",
  CET1240 "HERPESVIRUS":
O_W_NODULE;
  E_CFU1400 "DECREASED FUNCTION", R_NODULE:
O_C_CSYC3ZZ "EXTREMITY REGION NOS"
O_C_CSYC4ZZ "UPPER EXTREMITY, REGION"
O_C_CSYC5ZZ "LOWER EXTREMITY REGION";
  E_CFU2A85 "FEELING OF COLDNESS":
F_X C_CFUZZZZ "?";
  E_CET4250 "COLD":
N_;.
  E_CSY4000 "RESPIRATORY",
  CFU4400 "INFECTION",
  CET1284 "COMMON COLD"
IN
N_;.
  P_T_314
PAIN
N_;.
  E_CFU3000 "PAIN"
PAINFUL
A_PAIN
POST
F_X C_DTBZZZZ "?" &
O_C_CFUZZZZ "?";
  E_DT07500 "INDEFINITE TIME AFTER BASE":
O_C_DACZZZZ "ACTIVITY UNSPECIFIED"
  E_DT07500 "INDEFINITE TIME AFTER BASE";
  DTBA000 "SYSTEM (PHYSIOLOGIC FUNCTION)":
O_C_DTBZZZZ "?";
  E_DT07500 "INDEFINITE TIME AFTER BASE",
  CFUC100 "CHECKUP/FOLLOWUP":
!F_X C_CFUZZZZ "?" C_CFUEZZZ "SURGICAL PROCEDURES" &
F_X C_CFUZZZZ "?" C_CFUZZZZ "?"
  E_DT07500 "INDEFINITE TIME AFTER BASE",
  DTBB000 "FUNCTION":
F_X C_CFUZZZZ "?" &
F_Z C_CSYZZZ "PSYCHOSOCIAL"
  E_DQU0310 "STATUS POST",
O_C_CFUEZZZ "SURGICAL PROCEDURES"
  E_DQU0310 "STATUS POST":
N_;.
  E_DLTO040 "BEHIND (POSTERIOR)"

Figure 8-6. Sample Word Dictionary Entries
Figure 8-6 displays word dictionary definitions for the words COLD, IN, PAIN, PAINFUL, and POST. These definitions demonstrate almost all of the constructs available in the word dictionary. Each word sense meaning is separated by a colon. A word sense meaning is not necessarily the same as one would find in an ordinary dictionary, i.e., the word sense meanings for any word in the word dictionary are different from one another because they either instruct the encoder to assign different dimension codes and/or they require a different test before codes can be assigned. For example, the first word sense meaning for COLD is used if COLD is followed by SORE or SORES. The instruction portion of this word sense meaning for COLD instructs the encoder to remove the words SORE and SORES from any further consideration and assign the codes CSY6214-"Oral Commissure," CFU4400-"Infection," and CET1240-"Herpesvirus."

The second word sense meaning for COLD tests for COLD cooccurring with the word NODULE. This tests for COLD meaning no radioisotope up-take as in COLD NODULE THYROID. During construction of the word dictionary an attempt was made to balance generality and specificity, e.g., a cooccurrence test is more general than a format test so this second test on COLD would also produce a correct encoding of THYROID NODULE COLD. A format test is more restrictive than a cooccurrence test because it requires a specific word order. If the second test were met for COLD, the instruction portion would be used, i.e., the code CFU1400-"Decreased Function" would be assigned and the word NODULE would be removed.

The third word sense meaning for COLD is a feeling of coldness which only occurred in the CHCP problem-definitions in reference to an extremity region. Therefore, the test portion tests for the occurrence
of any System code associated with an extremity region. The "ZZ" on the end of the codes indicates that the test will be met if any code lower in the hierarchy is present. For example, CSYC460-"Arm Region" is lower in the hierarchy than CSYC400-"Upper Extremity, Region" so the problem-definition COLD ARM, would invoke this third word sense meaning for COLD.

The fourth word sense meaning for COLD is used whenever COLD means cold temperature. The format test specifies that any problem-definition having COLD followed by any word which produces a Function code should use this word sense meaning. The "ZZZZ" means any Function code is acceptable.

The last word sense meaning for COLD is used by default whenever none of the previous tests are met. It has the null test "N_" which tells the encoder to follow the instruction without performing any test. In this case the word sense meaning is that for the common cold.

The second word in Figure 8-6 is the preposition IN. The instruction portion of the definition tells the encoder to assign the word type, 314, to the word. No codes are assigned. The word type assignment is used by the rules section of the encoder.

The next two words in Figure 8-6, PAIN and PAINFUL are examples of the alternate forms usage. Whenever PAINFUL occurs in a problem-definition, the encoder uses the definition of PAIN.

POST has a lengthy entry in the word dictionary. The tests are used to disambiguate three different meanings for POST. The first meaning is that of posterior or spatially located behind. The second meaning is that of temporally after, and the third meaning is the rather vague meaning of status post or in a state occurring temporally after
some event. With one exception, the tests and instructions for POST are very similar to those of COLD.

The construct which is shown for POST but which does not occur with COLD is in the fourth word sense meaning of POST where the exclamation point indicates negation of a test. If POST is not followed by a Function code then a surgical procedure Function code but is followed by two function codes, then the fourth instruction is invoked. The purpose here is to skip the fourth word sense meaning with problem-definitions such as POST OPERATION MASTECTOMY but to use this word sense meaning with problem-definitions such as POST MASTECTOMY PAIN. The meaningful difference between these two examples is that POST OPERATION MASTECTOMY has the second word after POST coding into a surgical procedure, whereas this is not the case in POST MASTECTOMY PAIN. POST OPERATION MASTECTOMY invokes the sixth word sense meaning of POST, i.e., being in a state after a mastectomy, and POST MASTECTOMY PAIN invokes the fifth word sense meaning, i.e., pain following in time after a mastectomy.

Description of Rules File

After the word dictionary provides a potential set of dimension codes, the encoder enters the rules section which has three major functions.

(1) Further assignment or refinement of codes;
(2) Additional tests for total content;
(3) Arrangement of codes into their final form.

Each rule is composed of three parts; the form is:
\[ \text{<rule> ::= <pattern type><pattern><format>} \]

If <pattern> matches the set of codes, then the codes are transformed into <format>.

A <pattern> may specify the content of a code set with or without regard to order. In the first instance, <pattern type> is 0 and its use is similar to that in the word dictionary, i.e., <pattern> can be represented in the code set in any order for the rule to be followed. The only other <pattern type> is F. It specifies that <pattern> must match the code set in the prescribed order before the rule is invoked. An F was added to specify that the pattern must match the set of codes exactly, i.e., there could be no codes left unmatched.

A <pattern> has the following form:

\[ \text{<pattern>::= (<codep>^<rangep>^<typep>)[(^,&)<pattern>]} \]

Code, range, and type patterns—<codep>, <rangep>, and <typep>—are in the following form:

\[ \text{<codep>::=<dim><num>^<dim>(<num>)^<dim><num>} \]

\[ \text{<rangep>::=<dim>[<num>-<num>]^<dim>[[<num>-<num>]]^<dim>[<num>-<num>]} \]

\[ \text{<typep>::={"<type>"("<type>"}} \]
A <dim> is a three character alphabetic code for a specific dimension, e.g., CSY for System. A <num> is a four character hexadecimal code within a dimension. A "." within <num> matches any character. A <type> is a numeric code for special words and punctuation.

A <format> has the form:

<format>::=<dim><num>^<dim>#<digit>^#<digit><num>^#<digit>^<breaker>^<filler>^[,<format>]

or

<format>::=*1&F_

The <dim> and <num> are the same as in <pattern>. A <digit> is a single digit numeric, j, specifying the jth set of parentheses in the pattern. The characters within the code set which match the part of the pattern within the jth set of parentheses, are lifted out of the code set and inserted into <format> in place of #<j>. The <breaker> is a conjoining item, & or |, which separates <format> into more than one coded clinical event. A <filler>, F_, instructs the rule processor to fill-in anything not matched within parentheses from <pattern> into <format>. If the <format> is *1&F_, then everything to the left of the matched <pattern> and everything within parentheses is placed in <format>. The remainder of the code string is placed after the "."

A few examples of rules further illustrate their form. The
problem-definition, CHRONIC KNEE AND ANKLE PAIN, produces the following set of codes and word types from the word dictionary routines: DTD7500-"Long Duration Implied," CSYC586-"Knee, Region," 340-"Conjunction," CSYC4A0-"Ankle, Region," and CFU3000-"Pain." This set matches the <pattern> 

\[ \text{F}_{-}(\text{DTD}..., \text{CSY}..., ^{340}, \text{CSY}..., \text{CFU}...) \]

so the format

\[ >\#1,\#2,\#4&\#1,\#3,\#4 \]

instructs the encoder to change the form of the codes. Each number in the <format> corresponds to a set of parentheses in the <pattern>. Once a matching pattern is found, the rules section "picks-up" the codes within parentheses and places the codes in the positions specified in the <format>. In this example the word type, 340, for AND is not "picked-up." It is replaced by the conjoining symbol, "&," which is used to delimit conjoined coded clinical events. After the rule is applied, the codes are changed to DTD7500-"Long Duration Implied," CSYC486-"Knee, Region," CFU3000-"Pain," "&," DTD7500-"Long Duration Implied," CSYC5A0-"Ankle, Region," and CFU3000-"Pain."

The problem-definition, GLASS IN EYE, matches the <pattern> in the rule

\[ \text{F}_{-}(\text{CET}[4730-473F]), ^{314}, \text{CSY}..., >\text{CFU9D80}, \#1, \#2, \text{F}_{-}. \]
Codes produced by the word dictionary section are CET473F-"Glass Slivers," 314-"Preposition, in," and CSY2000-"Eye." The rule inserts the code CFU9D80-"Foreign Body." The "F_" is not used. It instructs the encoder to place anything remaining after a match into the position it holds in the <format>. The set of codes assigned to this problem-definition are: CFU9D80-"Foreign Body," CET473F-"Glass Slivers," and CSY2000-"Eye."

One rule which changes a System code to a Topography code is

\[
0_{CSY}(C...),(CSY....)>CTO#1,#2,F_.
\]

It is used for problem-definitions such as BURN R HAND. The word dictionary routines produce the following codes: CSYA000-"Skin," CFU9C20-"Burn," DLT1000-"Right," and CSYC4A0-"Hand." The System code "C4A0" for hand is "picked-up" and placed in the #1 position in the <format>. The complete code for skin is put in place of #2. DLT1000-"Right" and CFU9C20-"Burn" are put into the position of "F_." The final set of codes becomes CTOC4A0-"Hand," CSYA000-"Skin," CFU9C20-"Burn," and DLT1000-"Right."

The rules file "evolved" as needed during tuning with the design set. They became almost a grammar for problem-definitions using the dimension codes as semantic units. Figure 8-7 is an attempt to classify the 146 rules developed for automatically encoding the design set. The objective for 66 rules is to disjoin a set of codes into more than one coded clinical event. Another 45 rules involve changing some dimensions to others. Eighteen rules insert codes and 17 are used to ignore unneeded codes or word types.
<table>
<thead>
<tr>
<th>NUMBER OF RULES</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Ignore articles and words of certain types</td>
</tr>
<tr>
<td>2</td>
<td>Change the disease specified in an immunization to a drug</td>
</tr>
<tr>
<td>2</td>
<td>Assign codes to &quot;+&quot; and &quot;?&quot;</td>
</tr>
<tr>
<td>2</td>
<td>Change codes conjoined by &quot;+&quot;'s to multiple coded clinical events</td>
</tr>
<tr>
<td>1</td>
<td>Convert a preposition followed by an Activity code to a time code</td>
</tr>
<tr>
<td>1</td>
<td>Convert &quot;since&quot; followed by a number to a time code</td>
</tr>
<tr>
<td>4</td>
<td>Arrange a set of codes containing &quot;with&quot; into two coded clinical events</td>
</tr>
<tr>
<td>2</td>
<td>Change some uses of &quot;,&quot; and &quot;&amp;&quot; to conjoining indicators of coded clinical events</td>
</tr>
<tr>
<td>4</td>
<td>Insert a Function code when &quot;in&quot; occurs in a certain format to indicate either an ectopic substance or a foreign body</td>
</tr>
<tr>
<td>1</td>
<td>Disjoin two coded clinical events separated by a &quot;/&quot;</td>
</tr>
<tr>
<td>6</td>
<td>Ignore other words of different types</td>
</tr>
<tr>
<td>29</td>
<td>Convert codes with conjoining &quot;and&quot; to multiple coded clinical events</td>
</tr>
<tr>
<td>1</td>
<td>Disjoin coded clinical events conjoined by the phrase &quot;rule out&quot;</td>
</tr>
<tr>
<td>1</td>
<td>Ignore &quot;-&quot;</td>
</tr>
<tr>
<td>1</td>
<td>Make two separate coded clinical events linked by a Continuation code when CFU2401-&quot;Fear&quot; occurs with another Function code</td>
</tr>
<tr>
<td>10</td>
<td>Disjoin coded clinical events when one is the etiology of another</td>
</tr>
<tr>
<td>1</td>
<td>Disjoin the occurrence of three coded clinical events in one problem-definition</td>
</tr>
<tr>
<td>1</td>
<td>Ignore Function codes such as those for &quot;problem&quot; and &quot;disease&quot; when there is another Function code within the clinical event</td>
</tr>
<tr>
<td>5</td>
<td>Disjoin coded clinical events with procedure Function codes and other non-procedure Function codes</td>
</tr>
<tr>
<td>2</td>
<td>Disjoin and insert a Continuation link when the Time Base refers to a separate coded clinical event</td>
</tr>
</tbody>
</table>

Figure 8-7. Categorization of Rules
<table>
<thead>
<tr>
<th>NUMBER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Disjoin some common occurrences of multiple coded clinical events within one problem-definition which have no conjoining markers</td>
</tr>
<tr>
<td>1</td>
<td>Insert a Continuation code when Time dimension codes occur with a code for a surgical procedure</td>
</tr>
<tr>
<td>2</td>
<td>Make two coded clinical events, one with a Continuation code, when CFU9A70-&quot;Fistula&quot; occurs with two System codes</td>
</tr>
<tr>
<td>5</td>
<td>Change Function codes which are within the range of a Modifier dimension and occur with another Function code to Modifier codes</td>
</tr>
<tr>
<td>11</td>
<td>Change Function codes indicating &quot;infection,&quot; &quot;bleeding,&quot; &quot;pigmentation,&quot; &quot;rupture,&quot; or &quot;pain&quot; and occurring with another Function code to two coded clinical events linked by a Continuation code</td>
</tr>
<tr>
<td>2</td>
<td>Consider all the specific fingers and toes to be subordinate to the codes for non-specific fingers and toes</td>
</tr>
<tr>
<td>13</td>
<td>Change System codes which are in the range of Topography codes and have another System code within the coded clinical event to Topography codes</td>
</tr>
<tr>
<td>5</td>
<td>Insert the Function code, CFU9D80-&quot;Foreign Body&quot; when an Etiology code for small object only occurs with a System code</td>
</tr>
<tr>
<td>1</td>
<td>Insert a Function code, CFU4400-&quot;Infection&quot; when infective organisms occur without a Function code</td>
</tr>
<tr>
<td>1</td>
<td>Insert an Etiology code, CET6500-&quot;Drugs&quot; when a Drug code is followed by a Function code and there are no other dimension codes</td>
</tr>
<tr>
<td>4</td>
<td>Insert Event Type codes for procedures</td>
</tr>
<tr>
<td>1</td>
<td>Ignore the Activity code, DACF000-&quot;Physiologic Function&quot;</td>
</tr>
<tr>
<td>8</td>
<td>Manipulate Time, Quantity, and Units codes to achieve a concise representation</td>
</tr>
<tr>
<td>1</td>
<td>Remove any Event Type codes occurring with IET8500-&quot;History&quot;</td>
</tr>
</tbody>
</table>

Figure 8-7 (con't). Categorization of Rules
CHAPTER IX

RESULTS

Design Set Results

Figure 9-1 summarizes the final results of manually and automatically encoding the design set. There were a total of 5,465 unique problem-definitions in the design set. These were selected from the total CHCP unique problem-definitions in the manner described in Chapter VI, Methodology. Figure 9-1a is a tabulation of results for unique problem-definitions. Figure 9-1b is a tabulation of all the problem-definitions which these unique problem-definitions represented.

Table 9-1a may be explained as follows. The two columns represent a manual encoder's view of the unique problem-definitions in the design set. The manual encoder, in consultation with a medical adviser, considered 254 of 5,465 unique problem-definitions to be uncodeable.

Figure 9-2 displays a rather arbitrary categorization of the reasons some CHCP problem-definitions in the design set were manually uncodeable. Choosing some categories for problem-definitions involves guessing what might have happened or what might have been the intent when the problem-definitions were input into the computer. This is especially true of the spelling errors and the statements which might have valid meanings within specialty areas of medicine. In addition, some categories are overlapping. For example, illogical problem-definitions may be caused by spelling errors or keypunching errors.
<table>
<thead>
<tr>
<th></th>
<th>Manually Codeable</th>
<th>Manually Uncodeable</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Codeable</td>
<td>5196 (.951)</td>
<td>56 (.010)</td>
<td>5252 (.961)</td>
</tr>
<tr>
<td>Agreement</td>
<td>5191 (.999)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disagreement</td>
<td>5 (.001)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatically Uncodeable</td>
<td>15 (.003)</td>
<td>198 (.036)</td>
<td>213 (.039)</td>
</tr>
<tr>
<td>Dictionary Tests</td>
<td>0</td>
<td>8 (.040)</td>
<td></td>
</tr>
<tr>
<td>Undefined Words</td>
<td>0</td>
<td>67 (.338)</td>
<td></td>
</tr>
<tr>
<td>Dictionary Request</td>
<td>0</td>
<td>8 (.040)</td>
<td></td>
</tr>
<tr>
<td>Rule Request</td>
<td>15 (1.000)</td>
<td>115 (.581)</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>5211 (.954)</td>
<td>254 (.046)</td>
<td>5465</td>
</tr>
</tbody>
</table>

**Figure 9-1a.** Summary of Encoding Results for Unique Problem-definitions in the Design Set

131
<table>
<thead>
<tr>
<th></th>
<th>Manually Codeable</th>
<th>Manually Uncodeable</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Codeable</td>
<td>114086 (.986)</td>
<td>279 (.002)</td>
<td>114365 (.988)</td>
</tr>
<tr>
<td>Agreement</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>114075 (.999)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disagreement</td>
<td></td>
<td>11 (.00+)</td>
<td></td>
</tr>
<tr>
<td>Automatically Uncodeable</td>
<td>35 (.000+)</td>
<td>1307 (.011)</td>
<td>1342 (.012)</td>
</tr>
<tr>
<td>Dictionary Tests</td>
<td>0</td>
<td>314 (.240)</td>
<td></td>
</tr>
<tr>
<td>Undefined Words</td>
<td>0</td>
<td>438 (.335)</td>
<td></td>
</tr>
<tr>
<td>Dictionary Request</td>
<td>0</td>
<td>27 (.021)</td>
<td></td>
</tr>
<tr>
<td>Rule Request</td>
<td>35 (1.000)</td>
<td>528 (.404)</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>114121</td>
<td>1586 (.014)</td>
<td>115707</td>
</tr>
</tbody>
</table>

Figure 9-1b. Summary of Encoding Results for All Problem-definitions in the Design Set
<table>
<thead>
<tr>
<th>Reasons for Being Manually Uncodable</th>
<th>Total Problem-definitions</th>
<th>Unique Problem-definitions</th>
<th>Unique Problem-definitions with Undetected Errors</th>
<th>Unique Problems on Falling Dictionary Tests</th>
<th>Unique Problem-definitions with Undefined Words</th>
<th>Unique Problem-definitions which had Dictionary Requests for Information</th>
<th>Unique Problem-definitions which Failed in the Rules Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Content Beyond the Domain of Coding Scheme</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Contained is System But No Function</td>
<td>139</td>
<td>44</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>41</td>
</tr>
<tr>
<td>Only the Name of a Medication</td>
<td>96</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>Spelling Error which Could Possibly be Corrected from Context</td>
<td>74</td>
<td>17</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Spelling Error with No Context Useful for Correction</td>
<td>97</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unfamiliar terms which Might Have Some Meaning in a Speciality Area of Medicine</td>
<td>246</td>
<td>45</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Problem not Stated Clearly Enough to Encode</td>
<td>254</td>
<td>46</td>
<td>19</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>Possible Keypuncher Truncation or Error</td>
<td>159</td>
<td>23</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Specific Information Missing</td>
<td>25</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>No Statement of Problem -- Especially Function</td>
<td>418</td>
<td>15</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Illogical</td>
<td>74</td>
<td>25</td>
<td>14</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Totals</td>
<td>1586</td>
<td>254</td>
<td>56</td>
<td>0</td>
<td>67</td>
<td>8</td>
<td>115</td>
</tr>
</tbody>
</table>

Figure 9-2. Reasons for Considering Some Design Set Problem-definitions Manually Uncodeable
Although the classification is faulty, it does convey a few of the difficulties encountered in trying to interpret problem-definitions out-of-context, and several years after they have been recorded. In addition, the problem-definitions were interpreted through an intermediary, a keypuncher.

To further explain the problem-definitions considered manually un-codeable, a few examples from each category are necessary. In the first category, three unique problem-definitions contained information beyond the scope of ACS. These were GENERAL CARE EXECUTIVE ANNUAL CHECK, PERMIT SIGNED FOR STERILIZATION, and BACK PAIN WORKMAN'S COMP. It is conceivable that a medical coding scheme could code these problem-definitions; but a decision was made to ignore them in the present work.

There were many problem-definitions in the second category which had System dimensions, but no Function dimensions. A few of these are GALL BLADDER, R ELBOW, NOSE CARDIAC and CORNEA R EYE. These problem-definitions are incomplete problem statements and are not manually codeable.

In category three, only possible medications were given in the problem-definitions. Examples are IUD, PE TUBES, and CONTACT LENS. Without additional information indicating, for example, whether the medications produced problems or were administered to treat problems, these problem-definitions are manually uncodeable.

Category four contains problem-definitions with spelling errors which could possibly be corrected from context. Examples are INFECTIOUS MONTH, REFRACTION IRON, and RHEUMATIC HEAD. In these examples, MONTH is probably MONO, IRON is probably ERROR, and HEAD is probably HEART DISEASE.
Correcting these as special cases destroys the ability to link back to the original problem statement. Correcting these misspellings in the dictionary with a pointer to the correct word was not possible because the misspellings are legitimate words in their own right. Since problem-definitions with these types of spelling errors were few in number, choosing to ignore them by considering them manually uncodeable was sensible.

There were a few spelling errors with no context within the problem-definitions useful for spelling correction. These are classified in category five. Examples are TOREMIA, ANGRIA, and BARTOLOZZIS SYNDROME. Manually assigning a code, in these cases is impossible without knowledge of the content of the total medical record.

Several problem-definitions might have meaning within a specialty area but they are not on any list of standard medical abbreviations and they are unfamiliar to both this researcher and the medical consultant. However, they are either repeated several times or "look" as though they might have some legitimate medical meaning. Of course, they could also be spelling or keypunching errors. Quite a few were possible abbreviations, such as TAH BSO LYSIS OF ADHESION, LFD, LVED, RH AFTER OPERATION, and OB OBESITY.

Category seven contains problem-definitions lacking sufficient clarity to manually encode—especially without certain items from the rest of the medical record. For example, the problem-definition WEIGHT GAIN, could be interpreted as the patient "desires" to gain weight or the patient "is" gaining weight. A check of the physical exam with height, weight, sex, and age would explain this problem-definition.
Another problem, PREMARITAL DISEASE, leaves much information open to the imagination. The patient could have desired a premarital disease laboratory test or he/she could have taken the test, with positive results. Without integrating the automatic encoder into a fully automated medical record system, or at least being able to ask a user in real-time for additional information, these problem-definitions are manually uncodeable.

Category eight contains problem-definitions which appear to have uncorrectable keypunching errors or truncation. Examples are ABNORMAL R, SLAN RASH 2GROIN&UNDER R BREAST, and INFECTION COUGH FOR 1. Without the original medical record, these are impossible to manually encode.

Problem-definitions with specific missing information are grouped in category nine. Examples are CYSTIC DUCT, MEATAL ULCER, and HILAR MASS. They each contain structures which may occur in several places throughout the body. Additional words which could appear in this category are OPENING, ORIFICE, and CANAL. Each of these words define a "type" of structure, but the identity of the "specific" structure depends upon context. Most of these problem-definitions produced a request for additional information from the word dictionary, since the automatic encoder gives no meaning to these words used alone. Their meaning is stored with the word upon which they depend. For example, the dictionary entry for THYROGLOSSAL contains the meaning for THYROGLOSSAL DUCT just as EAR contains the meaning for EAR CANAL. In category nine, these "dependent" words do not appear with any word which could make their meaning specific. Therefore, the dictionary asks for information; and, since that information is not available, they are considered manually uncodeable.
Category ten contains problem-definitions which have no statement of a problem. For example, MENOPAUSE, MENSTRUAL BLEEDING, and MULTI-PARITY are all quite normal functions. One may surmise that there was some difficulty associated with these generally normal body states; however, even a manual encoder needs more facts before a code can be assigned. Within this grouping, there are also many statements which are simply comments, such as WORSE, ROUTINE, or DICTATION. Since there is no actual statement of a medical problem, these problem-definitions are manually uncodeable.

Category eleven contains problems which are medically illogical. For example, NIGHT NOSTRILS, LIBIDO RETICULARIS, CATARACT IN THROAT, and SCHLERODERMA OF BURSITIS. Without the ability to use some medical logic, the automatic encoder fails to recognize that, for example, cataracts do not occur in throats.

The present automatic encoder is not designed with the ability to recognize all illogical relationships among dimensions. The primary reason being the extreme difficulty of the task, which requires not only a logical representation of medical data but also a "learning machine" approach. Further discussion of this topic appears in Chapter XI, which examines recommendations for the future.

Having discussed, at length, the reasons for considering some problem-definitions in the design set not codeable, Figure 9-la will be further explained. The rows depict the automatic encoder's view of the design set. When the automatic encoder is run in automatic mode, and a set of manually assigned codes is available, it compares the manually assigned codes to the codes which it assigns. As shown in Figure 9-la,
a 99.9% agreement exists between automatically codeable and manually
codeable problem-definitions in the design set.

Figure 9-3 is a list of the five unique problem-definitions in the
design set which had disagreement between manual and automatically as­
signed codes. In the first problem-definition, COLD COUGH, a word
dictionary format test tells the encoder to use an Etiology code for
"cold temperature" whenever cold is followed by a word containing a
function. Therefore, it fails (in the absence of a conjoining word) to
disjoin COLD COUGH into two separate coded clinical events.

The next four problem-definitions in Figure 9-3 demonstrate the
difficulty in determining the meaning of the word POST. In the second
problem-definition, shown in Figure 9-3, the encoder failed to determine
that the operation was a herniorrhaphy. If the problem-definition had
been stated POST HERNIORRHAPHY CHECK, the encoder would have coded it
correctly. In the third problem-definition, the encoder viewed MYOCARDIAL
as a System word and encoded POST as POSTERIOR. The same assumption, i.e.,
POSTERIOR, would have been made in the fourth problem-definition in
Figure 9-3, except that POLYPECTOMY is a surgical procedure, and it
signals the encoder to select a time meaning for POST. However, the
information that BLEEDING follows in time after POLYPECTOMY is not dis­
cerned by the encoder. Finally, the encoder recognizes the cause-effect,
i.e., etiology relation between RHEUMATIC and MITRAL STENOSIS, but it
fails to use POST to code the time relationship between RHEUMATIC and
MITRAL STENOSIS.

Returning again to the discussion of Figure 9-la, there are 15
unique problem-definitions which were manually codeable, but the automatic
1. %2 COLD COUGH
2. %2 HERNIORRHAPHY POST OPERATION CHECK
3. %3 POST MYOCARDIAL INFRACTION
4. %2 POST RECTAL POLYPECTOMY BLEEDING
5. %2 POST RHEUMATIC MITRAL STENOSIS

Figure 9-3. Design Set Problem-Definitions Which Had Automatically Assigned Codes in Disagreement with Manually Assigned Codes
encoder was unable to code them. As discussed in Chapter VI, this type of error corresponds to a false-positive.

For the design set there were 0.3% unique false-positives. Figure 9-4 is a listing of the 15 unique problem-definitions in the design set which were manually codeable but the automatic encoder failed to code. In each one of these problem-definitions, except the seventh, the automatic encoder had difficulty disjoining the statement into two or more coded clinical events. For example, in the first problem-definition the S&O means that the operation also included removal of the fallopian tubes and ovaries.

In the seventh problem-definition shown in Figure 9-4, HALLUX means great toe, CSYC5C8, but PLANTAR WART implies both a System and a Topography, i.e., CSYA000-"Skin" and CT0C5CD-"Plantar Region." The automatic encoder produced two System codes which the rule directives rejected. Reordering the ACS hierarchy to subordinate CSYC5C8-"Great Toe" to CSYC5CD-"Plantar Region" requires a major revision in the coding scheme with depth greater than four in the hierarchy. Chapter XI discusses recommendations for coding scheme improvements needed to automatically produce a correct set of codes in this type of situation.

As shown in Figure 9-1a, the rules section of the automatic encoder asked for user assistance to encode 15 problem-definitions which were manually codeable. None of the other subcategories under Automatically Un-codeable have any problem-definitions listed beside them because the purpose of iterations with the design set was to tune the word dictionary and rules for optimum automatic encoder performance. Therefore, all word dictionary tests are successful in producing codes. The following
1. %2 S/P HYSTERECTOMY BILATERAL S&O
2. %3 S/P VAGOTOMY PYLOROPLASTY
3. %2 SHOULDER SIDE & CHEST PAIN
4. %3 VULVA VAGINITIS
5. %2 WEAKNESS R ARM-LEG
6. %2 LARYNGO TRACHEITIS
7. %2 PLANTAR WART R HALLUX
8. %3 POLYP R THIGH ? FIBROMA
9. %3 COLD SORE THROAT
10. %2 EMOTIONAL PROBLEM FATIGUE
11. %2 HEARING LOSS SEROUS OTITIS
12. %2 BILATERAL MYRINGOTOMY & PE TUBES
13. %4 CALCIFICATION AORTIC STENOSIS
14. %2 CANCER RECTUM AFTER COLOSTOMY
15. %2 CELLULITIS ABSCESS

Figure 9-4. Design Set Problem-Definitions Which Were Manually Codeable but Automatically Uncodeable
section discusses an attempt to quantify the value of the word dictionary tests in the automatic encoder.

**Estimating the Value of the Word Dictionary Tests**

To estimate the value of word dictionary tests, the automatic encoder was modified to output onto a separate file each word and the word sense meaning used in the 5,211 problem-definitions of the design set which coded successfully. After sorting, the most frequently used word sense meanings became entries in a new word dictionary with no format or cooccurrence tests.

The output from automatically encoding the design set produced the results in Figure 9-5. These results represent the best the encoder could produce if the word dictionary contained no tests but had the most frequent word sense meaning (used in the design set) for every word. Disagreement rises from 0.1% to 4.8% and the number of Automatically Uncodeable problem-definitions rise from 0.3% to 3.2%. Of course, achieving these results in practice, requires some excellent prior hypotheses about word sense usage.

**Estimating the Value of the Rules Section**

The next logical method for evaluating a portion of the automatic encoder was to disconnect the rules section. Figure 9-6 shows the effect of running the automatic encoder without the rules section using the design set. The percentage of Disagreement jumps from 0.1% to 33.7%. In addition the number of manually uncodeable problem-definitions which the automatic encoder can detect drops.

Figure 9-7 shows the results of running the automatic encoder
<table>
<thead>
<tr>
<th></th>
<th>Manually Codeable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Codeable</td>
<td>5047 (.954)</td>
</tr>
<tr>
<td>Agreement</td>
<td>4807 (.952)</td>
</tr>
<tr>
<td>Disagreement</td>
<td>240 (.048)</td>
</tr>
<tr>
<td>Automatically Uncodable</td>
<td>164 (.032)</td>
</tr>
<tr>
<td>Dictionary Tests</td>
<td>0</td>
</tr>
<tr>
<td>Undefined Words</td>
<td>0</td>
</tr>
<tr>
<td>Dictionary Request</td>
<td>0</td>
</tr>
<tr>
<td>Rule Request</td>
<td>164 (1.000)</td>
</tr>
<tr>
<td>Totals</td>
<td>5211</td>
</tr>
</tbody>
</table>

Figure 9-5a. Summary of Automatically Encoding the Manually Codeable Unique Design Set Problem-definitions Using only the Most Frequent Word Sense Meanings in the Word Dictionary
<table>
<thead>
<tr>
<th></th>
<th>Manually Codeable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Codeable</td>
<td>108926 (.954)</td>
</tr>
<tr>
<td>Agreement</td>
<td>104402 (.959)</td>
</tr>
<tr>
<td>Disagreement</td>
<td>4524 (.042)</td>
</tr>
<tr>
<td>Automatically Uncodeable</td>
<td>5195 (.046)</td>
</tr>
<tr>
<td>Dictionary Tests</td>
<td>0</td>
</tr>
<tr>
<td>Undefined Words</td>
<td>0</td>
</tr>
<tr>
<td>Dictionary Request</td>
<td>0</td>
</tr>
<tr>
<td>Rule Request</td>
<td>5195 (1.000)</td>
</tr>
<tr>
<td>Totals</td>
<td>114121</td>
</tr>
</tbody>
</table>

**Figure 9-5b.** Summary of Automatically Encoding the Manually Codeable Design Set Problem—definitions using only the Most Frequent Word Sense Meanings in the Word Dictionary
<table>
<thead>
<tr>
<th></th>
<th>Manually Codeable</th>
<th>Manually Uncodeable</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Codeable</td>
<td>5211 (.954)</td>
<td>171 (.031)</td>
<td>5382 (.985)</td>
</tr>
<tr>
<td>Agreement</td>
<td>3454 (.663)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disagreement</td>
<td>1757 (.337)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatically Uncodeable</td>
<td>0</td>
<td>83 (.015)</td>
<td>83 (.015)</td>
</tr>
<tr>
<td>Dictionary Tests</td>
<td>0</td>
<td>8 (.096)</td>
<td></td>
</tr>
<tr>
<td>Undefined Words</td>
<td>0</td>
<td>67 (.807)</td>
<td></td>
</tr>
<tr>
<td>Dictionary Requests</td>
<td>0</td>
<td>8 (.096)</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>5211 (.954)</td>
<td>254 (.046)</td>
<td>5465</td>
</tr>
</tbody>
</table>

Figure 9-6a. Summary of Automatically Encoding the Unique Design Set Problem—definitions with No Rule Directives
<table>
<thead>
<tr>
<th></th>
<th>Manually Codeable</th>
<th>Manually Uncodeable</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Codeable</td>
<td>114121 (.986)</td>
<td>807 (.007)</td>
<td>114928 (.993)</td>
</tr>
<tr>
<td>Agreement</td>
<td>91730 (.804)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disagreement</td>
<td>22391 (.196)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatically Uncodeable</td>
<td>0</td>
<td>779 (.007)</td>
<td>779 (.007)</td>
</tr>
<tr>
<td>Dictionary Tests</td>
<td>0</td>
<td>314 (.403)</td>
<td></td>
</tr>
<tr>
<td>Undefined Words</td>
<td>0</td>
<td>438 (.562)</td>
<td></td>
</tr>
<tr>
<td>Dictionary Requests</td>
<td>0</td>
<td>27 (.035)</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>114121 (.968)</td>
<td>11586 (.014)</td>
<td>115707</td>
</tr>
</tbody>
</table>

Figure 9-6b. Summary of Automatically Encoding the Design Set Problem—definitions with No Rule Directives
### Figure 9-7a.
Summary of Automatically Encoding the Manually Codeable Unique Problem-Definitions in the Design Set Using only the Most Frequent Word Sense Meanings in the Word Dictionary and No Rule Directives

<table>
<thead>
<tr>
<th></th>
<th>Manually Codeable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Codeable</td>
<td>5211</td>
</tr>
<tr>
<td>Agreement</td>
<td>3215 (.617)</td>
</tr>
<tr>
<td>Disagreement</td>
<td>1996 (.383)</td>
</tr>
</tbody>
</table>

### Figure 9-7b.
Summary of Automatically Encoding the Manually Codeable Problem-Definitions in the Design Set Using only the Most Frequent Word Sense Meanings in the Word Dictionary and No Rule Directives

<table>
<thead>
<tr>
<th></th>
<th>Manually Codeable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Codeable</td>
<td>114121</td>
</tr>
<tr>
<td>Agreement</td>
<td>87330 (.765)</td>
</tr>
<tr>
<td>Disagreement</td>
<td>26791 (.235)</td>
</tr>
</tbody>
</table>
with the word dictionary of most frequent word sense meanings and no rules section. Disagreement with manually assigned codes rises from that shown in Figure 9-6; however, the increase is on the order of 4% which is comparable to the difference between using dictionary tests and using no dictionary tests with the most frequent word sense meanings.

**Results from a Test Set**

There were 18 unique problem-definitions in the test set which could not be manually coded. Figure 9-8 lists these problem-definitions. Eleven of these are problem-definitions with only dates or comments and no information about a medical problem. The phrase, SIGN SYMPTOM COMPLEX, is in common use at Grady. It indicates that the patient had a set of signs and symptoms which the physician could not label at the time the problem was recorded. Three other unique Grady problem-definitions, had abbreviations, CXR, ICDB, and RPR, which were unfamiliar to the medical consultant and this author. CXR is an abbreviation for chest xray; ICDB stands for incomplete data base; and RPR means rapid plasma reagin. "Incomplete data base" is a legitimate medical problem in the Grady setting. It means that the patient's data base, which is a structured form specifying primarily signs and symptoms, has not been completed. CSR and RPR are diagnostic tests.

Figure 9-9 is a tabulation of results from the automatic encoder after the test set had been manually encoded. Approximately two-thirds of the test set problem-definitions were automatically encoded correctly. In addition, the automatic encoder correctly detected the 18 unique problem-definitions which could not be manually encoded. Changes in
<table>
<thead>
<tr>
<th></th>
<th>Manually Codeable</th>
<th>Manually Uncodeable</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Codeable</td>
<td>67 (.670)</td>
<td>0</td>
<td>67 (.670)</td>
</tr>
<tr>
<td>Agreement</td>
<td>66 (.985)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disagreement</td>
<td>1 (.015)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatically Uncodeable</td>
<td>15 (.150)</td>
<td>18 (.180)</td>
<td>33 (.330)</td>
</tr>
<tr>
<td>Dictionary Tests</td>
<td>1 (.067)</td>
<td>5 (.278)</td>
<td></td>
</tr>
<tr>
<td>Undefined Words</td>
<td>13 (.967)</td>
<td>13 (.722)</td>
<td></td>
</tr>
<tr>
<td>Dictionary Request</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Rule Request</td>
<td>1 (.067)</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>82 (.820)</td>
<td>18 (.180)</td>
<td>100</td>
</tr>
</tbody>
</table>

Figure 9-9a. Summary of Encoding Results for Unique Problem-definitions in the Test Set
<table>
<thead>
<tr>
<th></th>
<th>Manually Codeable</th>
<th>Manually Uncodeable</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatically Codeable</td>
<td>389 (.641)</td>
<td>0</td>
<td>389 (.641)</td>
</tr>
<tr>
<td>Agreement</td>
<td>386 (.992)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disagreement</td>
<td>3 (.008)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatically Uncodeable</td>
<td>122 (.201)</td>
<td>96 (.158)</td>
<td>218 (.359)</td>
</tr>
<tr>
<td>Dictionary Tests</td>
<td>7 (.057)</td>
<td>26 (.271)</td>
<td></td>
</tr>
<tr>
<td>Undefined Words</td>
<td>102 (.836)</td>
<td>70 (.729)</td>
<td></td>
</tr>
<tr>
<td>Dictionary Request</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Rule Request</td>
<td>13 (.107)</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>511 (.842)</td>
<td>96 (.158)</td>
<td>607</td>
</tr>
</tbody>
</table>

Figure 9-9b. Summary of Encoding Results for Problem-definitions in the Test Set
vocabulary were the major reason for inability to automatically encode the test set.

Figure 9-10 lists the test set problem-definitions which were manually codeable but which the automatic encoder either failed to code or coded in disagreement with a manually assigned code. The problem-definition, PSYCHIATRIC SYMPTOMS, was the only problem-definition to automatically encode in disagreement with a manual code. The automatic encoder failed to encode CORONARY ARTERIOSCLEROTIC HEART DISEASE because arteries and heart both produced System codes.

ACS subordinates CSY5230-"Coronary Arteries" under CSY5000-"Cardiovascular System" and CSY5700-"Arteries" has a separate System code. The automatic encoder was not trained to connect arteries with heart and only uses the System code CSY5230-"Coronary Arteries." This situation could be systematically remedied in the rules section. However, it emphasizes the difficulty in arranging the hierarchy of ACS for facilitating both automatic encoding and retrieval. Retrievals keying on heart should produce problems involving coronary arteries but there also must be a separate Arteries code to use when arteries are not in the heart.

The source of difficulty in automatically encoding the remaining problem-definitions listed in Figure 9-10, was the word dictionary. In one of these, DEGENERATIVE JOINT DISEASE, the word dictionary contained the word sense meaning for degenerative arthritis; therefore, DEGENERATIVE without ARTHRITIS was undefined. All the other problem-definitions had words not defined in the word dictionary.

SYSTEMIC ARTERIAL HYPERTENSION was the most frequent
Figure 9-10. List of Test Set Problem-Definitions
Which Were Manually Codeable but
Failed to Automatically Encode
Correctly
problem-definition in the test set. The word "hypertension," when used without modifiers has the meaning of "systemic" disease, in that it occurs throughout the body. "Hypertension" alone, also means "arterial hypertension"; venous hypertension is a completely different symptom found in such diagnoses as congestive heart failure. Therefore, both the words "systemic" and "arterial" are redundant when used with "hypertension." The automatic encoder recognized HYPERTENSION and ARTERIAL and coded the problem-definition as it would HYPERTENSION unmodified. However, it had not encountered the word SYSTEMIC in the design set, so it refused to code the problem-definition with an unidentified word, even though it produced the correct code. This was also the case with HISTORY OF SYSTEMIC ARTERIAL HYPERTENSION.

This chapter has discussed the design set and test set results. The following chapter presents conclusions which can be drawn from these results.
The first objective of this dissertation was to extend ACS to code the information in problem-definitions. Extending ACS meant determining the number, structure, and content of dimensions other than the original System and Function. Twenty-seven dimensions were used to code the problem-definitions studied. This number is not fixed because there were three concepts which did not fit into any dimensions. They were "workman's compensation," "executive" used to modify a checkup, and the administrative procedure of signing an operation permission form. The only conclusion as to the number of dimensions which can be drawn from the present work is that the number of dimensions is on the order of 27 for a complete information representation.

Most dimensions developed to code the CHCP problem-definitions retain the hierarchical structure previously established in the System and Function dimensions. One intent of a hierarchical structure is to facilitate single-key retrieval, but it is also useful for eliminating redundancies in the final set of dimension codes for any problem-definition. Moreover, the total dimensional structure of ACS facilitates detecting the completeness of a medical statement, e.g., the rule requiring all coded clinical events to have a Function code.

In two instances, a strict hierarchical structure was not appropriate. The first of these occurred with the Laterality dimension.
Laterality requires a special structure because each Laterality code is a symbol for a point defined by a spatial coordinate system. Examples of scales on each axis are:

(right, middle, left)
(upper, middle, lower)
(outside, central, within)
(above, central, below)
.behind, central, in front of)
(diffuse, regional)

Therefore, the Laterality dimension is actually the reduction of many spatial dimensions into one ACS dimension.

The second instance, where a strict hierarchy was not appropriate, occurs when a dimension has both an integer or real scale and an ordinal scale. The Quantity dimension and four of the Time dimensions have codes for both definite numbers and indefinite numbers. A conversion from integer to ordinal would lose information, whereas a conversion from ordinal would lose information, whereas a conversion from ordinal to integer would add information. Two separate scales are needed within these dimensions to accurately portray the information within a problem-definition.

The processes of expanding ACS, designing the automatic encoder, and assigning manually correct codes to the design set, definitely interacted. One concrete example of the automatic encoder's influence upon ACS was that the encoder's use of the hierarchical structure to eliminate redundancies often dictated the position of a concept within
a dimension's hierarchy. The automatic encoder influenced assignment of manually correct codes by necessitating explicit logical links between multiple coded clinical events representative of one problem statement. In addition, the automatic encoder's need for an algorithmic way to recognize when a problem-definition was manually uncodeable influenced ACS development. Many other interactions were more subtle, and are not easily quantified. The total approach encourages interaction and introduces bias into all design aspects. However, it also forces coordination of all components for optimum performance; and the success achieved here contributes to the conclusion that the method is useful.

**Mechanisms for Automatic Encoding**

One question posed at the onset of this research was whether the complexity inherent in Natural Language problem-definitions would reflect in an automatic encoder. With some qualification, the conclusion reached here is no. An automatic encoder for problem-definitions need not be overly complex. In fact, the automatic encoder requires very little knowledge of the syntactic, or surface structure of problem-definitions. For example, recognition of adjectival markers in DIABETIC NEUROPATHY and ABDOMINAL PAIN, are useless when these problem-definitions have exactly the same codes as DIABETES NEUROPATHY and ABDOMEN PAIN.

Recognition of some prepositions is useful for automatic encoding e.g., TRAUMA-FINGER IN R EYE, S/P MASTECTOMY FOR CANCER, SORE ON NOSE. In the first problem-definition, "in" specifies that a finger was the agent of injury. In the second, "for" indicates that the following word or words are the cause of the surgery. The presence of the preposition
"on" in the third example, determines the meaning of "sore." However, in ABRASION ON NOSE, ABSCESS OF BUTTOCK, ALLERGY TO PENICILLIN, CHECK ON DIABETES, EXPOSURE TO PINWORMS, and STREPTOCOCCUS IN FAMILY, the preposition may be ignored. The automatic encoder presently distinguishes only four specific prepositions for the following purposes:

(1) "with" is sometimes used as a conjunction, as in COLD WITH COUGH.
(2) "on" is sometimes used to designate time, as in DYSPNEA ON EXERTION.
(3) "since" indicates time.
(4) "in" indicates location when it occurs within a certain complex.

Constraints on content and assumed knowledge of medical logic relieve the need for rigorous syntactic structure in problem-definitions which are written by physicians for themselves or other physicians to read. In addition, a "shorthand" style often ignores common surface structure markers. The conclusion is that much laborious surface structure processing is unnecessary for automatically encoding problem-definitions. However, the contribution made by an adequate coding scheme is unmeasured.

Word dictionary tests are primarily intended to disambiguate word sense meaning. Of the 3,686 words in the word dictionary with definitions containing codes, (this figure does not include the misspelling or alternate forms), 236 had cooccurrence tests, and 153, had format tests. A format test considers neighborhood word order, whereas a cooccurrence
test does not. Only 286 words had more than one word sense meaning indicated in the word dictionary. There were an average 1.2 word sense meanings per word in the design set. Figure 10-1 shows the number of word sense meanings by frequency in the design set. There are more words in the word dictionary than are in the design set because some words were defined from the word frequency list of all CHCP problem-definitions.

The rule directives are essential to the automatic encoder. When the rule directives were omitted, disagreement with manually assigned codes jumped from 0.1% to 33.7% for unique problem-definitions in the design set. Explanations may be found in the roles played by the rule directives and the diversity of form and content within problem-definitions. There were 522 problem-definitions with Event Type codes other than IET8000-"Problem." In each of these problem-definitions, the rules section inserted an Event Type code; therefore, there had to be at least 522 more failures occurring when the encoder was run without the rules section.

There were 589 problem-definitions which coded into more than one coded clinical event. Formation of multiple coded clinical events is performed solely in the rules section. Forty-five percent of the rules stored on the rules file are concerned with disjoining a set of codes into multiple coded clinical events. If these definite failures are added to the errors produced from failure to attach Event Type codes, the total, easily explainable, failures without rules is on the order of a thousand. Clearly, the rules contribute in other ways to successful automatic encoding.

The rules also direct the automatic encoder to ignore certain
Figure 10-1a. Number of Words in Design Set by Word Frequency
Figure 10-1b. Average Number of Word Sense Meanings in the Word Dictionary by Frequency of Words in the Design Set
words. They create Topography dimension codes, and all codes within the
group of Modifier dimensions. They also eliminate redundancies in the
final set of codes produced. The rules section is definitely beneficial
to the automatic encoding process.

**Feasibility of Automatic Encoding**

The last objective of this research was to demonstrate the feasibil­
ity of automatically encoding problem-definitions into ACS in a real­
time clinical environment. Any final conclusions are contingent upon
many factors not investigated within this research. For example, a data
base management system, a decoder, and a receptive clinical staff. How­
ever, considering this research as a pilot project, the preliminary re­
results indicate that real-time automatic encoding of Natural Language
problem-definitions is not only feasible in a clinical setting, but
needed for quality control.

The automatic encoder designed and implemented in this investi­
gation operates in real-time on a medium sized mini-computer. In
automatic mode, it presently encodes and performs a comparison with
manual codes at the approximate rate of 1000 problem-definitions per
hour. Response delay in manual mode is "tolerable" for a person ex­
tremely familiar with the problem-definitions being coded and the
operation of the encoder. Manual mode processing rates are largely de­
pendent upon the user's time to peruse information displayed on a
terminal. No special attention was paid to achieving efficient program­
ming code; therefore, it is safe to assume that program optimization
would greatly increase computation speed. Hardware costs and response
time are definitely not prohibitive for automatic encoding.

The accuracy demanded of an automatic encoder is an open question, i.e., what error percentage is acceptable to the medical community? Howell (1971) reported manual encoding errors as high as 17%. This would clearly be unacceptable in an automatic encoder. On the design set, the 0.1% disagreement rate would probably be very acceptable for aggregate statistics. However, even this small error would be intolerable for everyday clinical use.

The automatic encoder operating in real-time has one major advantage over manual encoding--errors can be corrected immediately. A person close to the source of information can quickly check the resulting codes; or, by using a decoder, a Natural Language paraphrase of the original statement. If this procedure is followed, a 0.1% error rate is well within reason.

From all perspectives, the error rate on the design set is lower than the error rate on the test set. However, the percentage of disagreement with manual codes is only 1.5% for unique problem-definitions in the test set and 0.8% for total problem-definitions in the test set. In addition, the automatic encoder correctly identifies some difficulty with all the problem-definitions in the test set which were manually uncodeable. In other words, in only one unique problem-definition out of 100 in the test set did the automatic encoder fail to recognize that it had made a mistake.

The success achieved with the test set is especially surprising when one considers that the patient population is demographically different from the design set and the physicians writing the problem
statements were completely different. Of course, the results may also be interpreted to indicate that the ambulatory nature of the patient population was the overriding reason for similarity between problem-definition content. Without speculating about reasons for success with the test set, overall results of this study indicate that content does not vary as much as has been previously postulated. Gross surface structure may vary, e.g., arrows in the Grady problem-definitions and standard comments in the CHCP problem-definition, but these are easily handled by current processing techniques.

Results with the test set in no way indicate that the automatic encoder designed in this research is completely general. Moreover, the statement that 67% correct automatic encoding in the test set is acceptable, may not be made. The conclusion to be drawn from the test set results is that techniques used in the automatic encoder show sufficient generality to be applicable and extendable to multiple clinical environments.

The decision about whether to use an automatic encoder is not a straight line, "yes or no" proposition. Some weight must also be placed upon quality control benefits. A panel of experts could not manually code some statements specified as medical problems in both the design set and the test set. In addition, abbreviations and word usage vary from one group of physicians to another. Success with the POMR hinges upon the concise statement of a medical problem. Not only the benefits from, but the application of a computerized medical record, depends, to a large extent, upon error free data input. Automatic encoding makes data input not only more palatable, but it also guarantees consistency and better
quality control.

One final conclusion has been reached through this work. It is never acceptable to draw general conclusions from one subject, but conviction behind this conclusion necessitates an exception. This research provided personal confirmation of Myers and Hendrickson's (1973) statement, i.e., manual encoding is a tedious task and requires more memory capability than a human possesses.
CHAPTER XI

RECOMMENDATIONS

Improvements

The design of the automatic encoder implemented in this research has room for improvement. On a general level, the encoder needs to be able to make use of some medical logic, i.e., it needs to recognize that cataracts do not occur in throats. Incorporating "learning machine" techniques into the automatic encoder can enable the encoder to self-acquire much medical logic.

As indicated previously, some of the dimensions used in this research need further development. In particular, the Medication dimension was left undefined. One other improvement to ACS is also needed. Current depth in the hierarchy is four with a range within levels of 16. In some specific places identified by this research, the depth needs to be increased. For example, Figure 11-1a, displays a portion of CSYCO00-"Body Regions" in the System dimension. A depth of four is clearly inadequate to reflect the inherent hierarchical organization. Figure 11-1b shows an improved organization with a depth of six.

Some minor detailed improvements are recommended for actual automatic encoder implementation. To increase computation speed, some technique, other than the current method of sequential search, needs to be used to access the rules file. In addition, Laterality should be incorporated within any rules pattern testing System or Topography. This
CSYC000 Body Regions
CSYC300 Extremity Region
CSYC400 Upper Extremity Region
CSYC4A0 Hand Region
   CSYC4A3 Finger Region
   CSYC4A4 Forefinger Region
   CSYC4AC Palm Region

Figure 11-la. Portions of the Current System Dimension Code with a Hierarchical Depth of Four

Body Regions
Extremity Region
   Upper Extremity Region
      Hand Region
         Palm Region
         Finger Region
            Forefinger Region

Figure 11-lb. Portions of an Improved System Dimension Code with a Hierarchical Depth of Six
would eliminate the need for variations of basic rule pattern statements whenever a System or Topography code occurs with a Laterality code. In the word dictionary, the format test should be able to specify a test involving words within a certain distance from the word being tested. The above improvements are recommended for both convenience and speed. In general, the entire encoder needs optimization. It currently accomplishes the intent of its design, but several considerations added late within the development phase make this prototype less than esthetically pleasing from a computer science point of view.

**Implications for Future Work**

Results achieved in this work direct the implementation of the automatic encoder within a working clinical environment. Within this setting, user interaction may be adequately studied. The present study only points out some possibilities for user interaction. In particular, user satisfaction and general applicability may be quantified.

One overall implication of the results achieved in this dissertation applies beyond the domain of medicine. Researchers in many different fields are currently struggling with ways to input and represent data from their respective areas in a computer. The development of ACS and the automatic encoder provides advice on methodology. First develop a coding scheme which not only meets specific retrieval needs, but captures as much explicit content as possible. Use a large set of representative data to structure the coding scheme and design an automatic encoder for encoding Natural Language statements of the data. If results with the design set are satisfactory, proceed, using the knowledge gained, to actualize the encoder, using the coding scheme, in a working environment.
APPENDIX

LISTING OF TEST SET PROBLEM-DEFINITIONS

59 SYSTEMIC ARTERIAL HYPERTENSION
41 ANEMIA
23 ADULT ONSET DIABETES MELLITUS
22 INCOMPLETE DATA BASE
21 OBESITY
19 CHEST PAIN
13 CORONARY ATHEROSCLEROTIC HEART DISEASE
13 TOBACCO ABUSE
10 EXOGENOUS OBESITY
10 HYPERTENSION
9 1/2/34
9 HYPERURICEMIA
8 ESSENTIAL HYPERTENSION
8 PROTEINURIA
8 WEIGHT LOSS
7 ABDOMINAL PAIN
7 CHRONIC ETHANOL ABUSE
7 CHRONIC BRONCHITIS
7 CHRONIC OBSTRUCTIVE PULMONARY DISEASE
7 DEGENERATIVE JOINT DISEASE
7 DIABETES MELLITUS
7 ETHANOL ABUSE
7 PERIPHERAL VASCULAR DISEASE
7 RENAL INSUFFICIENCY
6 1/2/34 NOT PRESENT
6 1/34
6 ALCOHOL ABUSE
6 CHRONIC ALCOHOL ABUSE
6 DIABETES MELLITUS, ADULT ONSET
6 PENICILLIN ALLERGY
6 PYURIA
6 SAH
6 SYMPTOM SIGN COMPLEX
5 ABNORMAL CXR
5 ABNORMAL EKG
5 BILATERAL CATARACTS
5 CIGARETTE ABUSE
5 NEUROLOGIC SYMPTOM SIGN COMPLEX
5 SIGN SYMPTOM COMPLEX
5 SIGN/SYMPTOM COMPLEX
5 SYSTOLIC MURMUR
5 RESOLVED
4 1934
CHRONIC PANCREATITIS
CHRONIC RENAL INSUFFICIENCY
DERMATOLOGIC PROBLEMS
DIABETES MELLITUS
DIVERTICULOSIS
REFERENCES


Gordon, B. L. (1973). Terminology and Coding of Medical Care Data. Medical Care Supplement, 11(2), 96-100.


Martha Carol Snyder Hansard was born in Kingsport, Tennessee on January 9, 1947. She graduated from Dobyns-Bennett High School in Kingsport in 1965, and earned a Bachelor of Science degree from the University of Tennessee in Knoxville in 1969. Her undergraduate major was Mathematics and her minor was Physics.

From 1969 to 1970, Ms. Hansard was employed as a Junior Statistical Quality Control Analyst by Univac in Bristol, Tennessee. The exposure to computers at Univac prompted her to begin graduate school in Computer Science at the University of Tennessee in Knoxville in 1970. As a full-time graduate student, she taught courses in programming and worked in a statistical applications group.

In 1971, she became employed full-time by Oak Ridge Associated Universities (ORAU) in Oak Ridge, Tennessee. While employed in the Medical Division of ORAU, she completed a Master's thesis dealing with computerized recognition of electrocardiogram waveforms. In 1973, she received a Master of Science degree.

Ms. Hansard, in 1975, entered the Biomedical Program of the School of Information and Computer Science at the Georgia Institute of Technology. While at Georgia Tech, she received support from the National Library of Medicine. In 1978, she was awarded the Doctor of Philosophy degree in Information and Computer Science from Georgia Tech and the Master of Science degree in Medical Science from Emory University in Atlanta, Georgia.
Ms. Hansard is currently employed at the ORAU Medical Division where she is a member of an epidemiology group studying long-term health hazards of energy related jobs.