

**ICODEX**

**A COMPUTER ASSISTED CODING TOOL OF THE  
INTERNATIONAL CLASSIFICATION OF DISEASES – VERSION 10  
FOR MORTALITY DATA**

**BY**

**JESUS PEINADO RODRIGUEZ**

**A thesis presented to the  
DEPARTMENT OF MEDICAL EDUCATION  
AND BIOMEDICAL INFORMATICS  
UNIVERSITY OF WASHINGTON**

**Submitted in partial fulfillment of the  
Requirements for the degree of  
Master in Biomedical and Health Informatics**

**Seattle - Washington**

**June 2003**

© Copyright by

**Jesus Peinado Rodriguez**

**2003**

## **ACKNOWLEDGEMENT**

I would like to thank my research advisors, Professors David Masuda and John Gennari for their constant guidance, endless patience, and constructive criticism throughout this project. It has been a pleasure working with them.

I am grateful to the members of the Boeing, Keith Butler and Gary Coen; to the members of the International Research and Training Program in Health Informatics, Ann Marie Kimball, Alicia Silva and Nedra Pautler; to the Biomedical and Health Informatics faculties with special emphasis to Professor Ira Kalet; to my colleagues Adriel Olortegui and Richard Phillips, and to my Peruvian and American friends for taking time from their busy schedules to support this effort.

I want to dedicate this thesis to my family and especially to my girlfriend, Nancy, for being so understanding of my long working research.

## Table of contents

Abstract.....	iv
<b>Chapter 1. Introduction.....</b>	<b>1</b>
<b>1.1 Introduction.....</b>	<b>1</b>
<b>1.2 Outline of this document.....</b>	<b>2</b>
<b>1.3 Organization of the document.....</b>	<b>2</b>
<b>Chapter 2. Review of relevant research.....</b>	<b>4</b>
<b>2.1 Problems with medical language.....</b>	<b>4</b>
<b>2.1.1 The process of the medical language standardization.....</b>	<b>4</b>
<b>2.1.2 Medical language.....</b>	<b>6</b>
2.1.2.1 The structural complexity of the medical language.....	6
2.1.2.2 Medical word parts with meaning.....	7
2.1.2.3 Medical word parts without meaning.....	7
<b>2.1.3 Controlled medical vocabularies.....</b>	<b>8</b>
2.1.3.1 Importance of controlled medical vocabularies in health care.....	8
2.1.3.2 Medical concepts in controlled vocabularies.....	9
2.1.3.3 Relationships and hierarchies in controlled vocabularies.....	10
<b>2.1.4 The International Classification of Diseases – ICD.....</b>	<b>12</b>
2.1.4.1 The ICD history.....	12
2.1.4.2 The ICD role.....	12
2.1.4.3 The ICD-10.....	12
<b>2.2 Problems with data entry and human errors.....</b>	<b>13</b>
<b>2.2.1 Data entry.....</b>	<b>13</b>
2.2.1.1 Goal of data entry and coding.....	13
2.2.1.2 Components of a coding system.....	13
2.2.1.3 The coding workflow.....	14
2.2.1.4 The problem with coding data.....	14
2.2.1.5 One of the solutions – human coders.....	14
2.2.1.6 Other problems with coding data.....	15
<b>2.2.2 Human errors.....</b>	<b>15</b>
2.2.2.1 Manual coding.....	15
2.2.2.2 Suggested schemas for detecting human errors.....	16
<b>2.2.3 Errors coding health data.....</b>	<b>16</b>
2.2.3.1 Errors coding mortality data.....	16
2.2.3.2 Errors in pragmatic relationships.....	17
2.2.3.3 Errors in cause of death.....	17
<b>2.2.4 Coding health software.....</b>	<b>18</b>
2.2.4.1 Coding software.....	18
2.2.4.2 Automated coding of mortality data.....	19
<b>2.3 Problems with vital statistics in Peru.....</b>	<b>21</b>
<b>2.3.1 Vital statistics in Peru.....</b>	<b>21</b>
2.3.1.1 Workflow of the death certificates in Peru.....	21

2.3.1.2 Errors in mortality data using the ICD-10 in Peru.....	22
<b>Chapter 3. The research approach and framework.....</b>	<b>26</b>
<b>3.1 Aims.....</b>	<b>26</b>
<b>3.2 Methodology.....</b>	<b>26</b>
<b>3.2.1 Intended basic functions for an accurate coding.....</b>	<b>26</b>
<b>3.2.2 Planed main tasks.....</b>	<b>27</b>
<b>Chapter 4. IcodeX system description.....</b>	<b>30</b>
<b>4.1 The ICD Lexicon.....</b>	<b>30</b>
4.1.1 The ICDL structure.....	30
4.1.2 The decision matrix table.....	32
4.1.3 Improving methods of access to the ICDL.....	33
<b>4.2 The semi-speller.....</b>	<b>34</b>
4.2.1 Semi-speller methods.....	35
4.2.2 Algorithm for comparing two words.....	37
4.2.3 Assignment of error values.....	39
4.2.4 Semi speller optimization methods.....	41
<b>4.3 ICD Analyzer.....</b>	<b>42</b>
4.3.1 Decision matrix analysis.....	42
4.3.2 Cause of death phrase analysis.....	43
4.3.3 ICD coding.....	43
<b>4.4 IcodeX in action.....</b>	<b>44</b>
4.4.1 Cooperative process in the cause of death coding.....	44
4.4.2 Basic and complex coding – Two main contexts.....	45
4.4.3 Coding rules of frequency.....	46
<b>Chapter 5. Summary of contributions and future research directions.....</b>	<b>48</b>
<b>5.1 Summary of conclusions.....</b>	<b>48</b>
<b>5.2 Future research directions.....</b>	<b>49</b>
<b>6. Bibliography.....</b>	<b>51</b>
<b>7. Appendices.....</b>	<b>55</b>
<b>A - IcodeX user Manual.....</b>	<b>55</b>
A.1 System requirements.....	55
A.2 Installation of the prototype.....	55
A.3 Using IcodeX.....	55
A.4 Known bugs.....	55
<b>B - The IcodeX interface.....</b>	<b>57</b>
<b>C - Making IcodeX.....</b>	<b>62</b>
C.1 Conceptualization phase.....	62
C.2 Programming and testing phase.....	64
<b>D - The IcodeX system.....</b>	<b>66</b>
<b>E - The list of abbreviations.....</b>	<b>68</b>

### List of figures

2.1.1 Entity schema about relationships among medical concepts.....	5
2.1.2.1 Structure of a complex medical term.....	7

2.1.3.2 Comparison of controlled vocabularies naming and identifying medical concepts	10
2.1.3.3 Comparison of controlled vocabularies by relationships and hierarchy	11
2.2.1 Health coding workflow for diagnosis	14
2.3.1.1 The death certificates workflow in Peru	22
3.2.2 Control of errors among sex, age and the cause of death	27
3.2.2 Working with the cause of death	28
4 The IcodeX system description	30
4.1 Hierarchical structure of the ICD-10	32
4.1.3 Size of the partitions of the ICD information	34
4.2.1 Schema about the semi-spelling correction method	36
4.2.2 Pseudo code of the auto completion function implemented in LISP	38
4.2.4 Representation of semi spelling correction	42
4.4.2 Simple and complex coding solved by the coding tool	46
B.1 Starting IcodeX	56
B.2 Coding the unity of age	56
B.3 Coding the age in numbers	56
B.4 Coding the sex	57
B.5 Coding the cause of death and optional functions	57
B.6 Coding the cause of death – selecting the category of death	58
B.7 Expanding the cause of death selection	58
B.8 Coding the specific cause of death	59
B.9 Selecting another database	59
B.10 Sequence control and range of age	60
C.1 Making IcodeX, schema of the project	61

### List of tables

Table 2.1.3.2 Inconsistencies in the automated translation between two terminologies	11
Table 2.2.3.1 Errors coding from health data to health codes	17
Table 2.3.1.2 Coding errors using ICD-10 - Ministry of Health from Peru	23
Table 4.1.1 Three-digit and four-digit codes of the ICD structure	31
Table 4.1.2 Structure of the decision matrix table	33
Table 4.2.3 General spelling errors categories by our support	40
Table 4.3.2 Relationship between the category of disease and the specific diseases	43
Table C.1 List of variables used in the analysis for mortality data from Peru	63

## **ABSTRACT**

Mortality is a key consideration in profiling population health status. The quality and accuracy of mortality data may significantly affect the quality of public health policy formulation. In developing countries such as Peru, there is a separation between public health policy and health statistics. Health statistics in Peru suffer from problems including difficulty in collecting and analyzing mortality data, the timeliness of mortality information, and mortality coding errors.

My project is aimed at mortality coding errors. The main objective of my research is to build a reliable and fast coding tool for coding basic cause of death with accurate ICD-10 codes.

My analysis of coding errors in Peru, along with a broader review of problems related to the coding of health data, has led me to develop a prototype system, named IcodeX, which I hope represents a first step toward a solution of these problems.

If used in Peru, my IcodeX tool will improve the accuracy of coding of cause of death. Better accuracy, if appropriately used, can lead to better health policy decisions.

The next step with IcodeX is an evaluation in the real world. My ultimate goal is to share my experiences and solutions with other Latin American countries with the same problem.

## CHAPTER 1

### 1. Introduction

#### 1.1 Introduction

The quality and accuracy of age specific mortality rates by cause are key considerations in profiling population health status. Also, there are directly related to the quality of making health policy because mortality is very important in vital statistics.

One important objective of information systems (IS) from the public health perspective is that IS should improve health care quality and population health. [Yasnof et al., 2001] In this sense, improving the quality of mortality data and making it useful and widely available to governmental health agencies, I aid to Peruvian public health policymakers in order to define needs for and effects of various welfare programs.

The development of correct public health policies is critically dependent on the use and analysis of health data. In developing countries there is a separation between public health policy and health statistics and this is true for Peru. As an example, In Peru there were incomparable questionnaires used in Peruvian public health surveys [Guzman et al., 1996]; the Peruvian outbreaks of emerging infection diseases were reported late [PAHO, 1999] and the statistical health information was inaccurate [Chirinos et al., 1994].

In the late 1980's, the Pan American Health Organization (PAHO) shifted from its traditional bimonthly morbidity reporting system to a focus on creating an accurate mortality database for member countries. Peru is member of PAHO and suffers from many of the same problems as other developing countries in collecting and analyzing mortality data. Basically, there are problems with inaccurate mortality information because there are coding errors of underlying causes of death. [Pan American Health Organization, 1996]

As we know, mortality information is generated by recording and analyzing mortality data. This project aims to reduce mortality coding errors that occur when a death certificate is coded for its subsequent analysis.

With this in mind I introduce a coding tool, named IcodeX, to help Peruvian coding users to use ICD-10 for mortality data. In fact, IcodeX not only solves the errors in the data entry process in Peru, but it will also help Peruvian health policy makers. As I focus on medical vocabularies, mortality coding rules and Peruvian coding errors, this initiative will contribute to a sustainable solution for developing countries.



Having studied common problems with coding errors in the health industry, I note that there are several flaws caused by its complexity. The complexity of health information is influenced by several elements in our systems that undermine, rather than support, recording of health information. Additionally, there are factors that interfere with the accuracy of information such as federal regulations, security issues and technological constraints. As a result, all of these factors affect the accurate capture of health information. [Slee et al., 2000]

In this way, I expect that IcodeX can work very well in a governmental health framework of a developing country like Peru that has a high rate of coding errors with mortality data. Indeed, I hope to help the Ministry of Health from Peru, one of the main actors among health policy makers in Peru.

I believe that the Ministry of health would be our end user. Therefore, this project is focused on the governmental health framework, because I work with the active collaboration of the Pan American Health Organization and the Ministry of Health from Peru.

## **1.2 Outline of this thesis**

Our previous analysis of coding errors in Peru, the review of relevant bibliography about human errors, medical vocabulary, and problems with data entry help us to understand the problems with mortality data; subsequently, with all these considerations I have implemented a modular solution. The use of modular elements such as a parsing analyzer, decision matrix searcher, syntactic and semantic checks permit us to detect input errors writing the underlying cause of death.

Another goal of this research is to show that it is possible to create a versatile tool for any kind of language. In fact, the organization of IcodeX allow us add more codes, different languages, and coding rules.

## **1.3 Organization of the thesis**

Chapter II contains a review of the relevant literature. An overview of parsing methods follows with a description of the current problems with mortality and coding health data. I also provide a brief overview of the medical language complexity, controlled vocabularies and problems with the human coding.

Chapter III describes the methodology and analysis process of the mortality data from Peru. I present a detailed description of the steps of conceptualizing, programming and testing.

Chapter IV covers the lexicon structure, and the semi-spelling methods. This chapter describes in detail the implemented algorithms that control the coding process. In the same way, examples of basic and useful functions are illustrated together with a relevant bibliographic review.

Chapter V summarizes the research and suggests the directions for future studies.

## CHAPTER 2

### 2. Review of relevant related research

In this chapter I review the relevant literature and current research on medical language, data entry, human errors, automated coding and the current state of vital statistics errors from Peru.

An overview of the involved problems on coding health data is succinctly described. I have organized this chapter in three parts; the first part is about the problems with the medical language, the second one describes the data entry problems and the third part is a descriptive analysis about coding errors in Peru.

#### 2.1 Problems with medical language

##### 2.1.1 The process of the medical language standardization

The standardization of the language in the medical field needs no longer be justified and we know that health data needs to be converted in a standardized knowledge or a formal language for solving problems such as medical language processing, medical information indexing and health ontology development. [Pacak et al., 1980; Wolff, 1987]

The standardization process has problems associated to the issues of the best health knowledge representation because of the complex association among meaning, sign, and symbol. Indeed, standardization of medical knowledge for numeric representation represents a challenge for medical language processing, which together with the maintenance of multiple classifications necessitates identifying consistent hierarchical placement and equivalence between concepts. [Brown et al., 1999]

The lack of a standard for changing representation makes it difficult to make updates for mapped or merged terminologies. In this sense, the coordination between the standard and the local terminologies may be essential for taking care of patients at the local level. [Gruber, 1993]

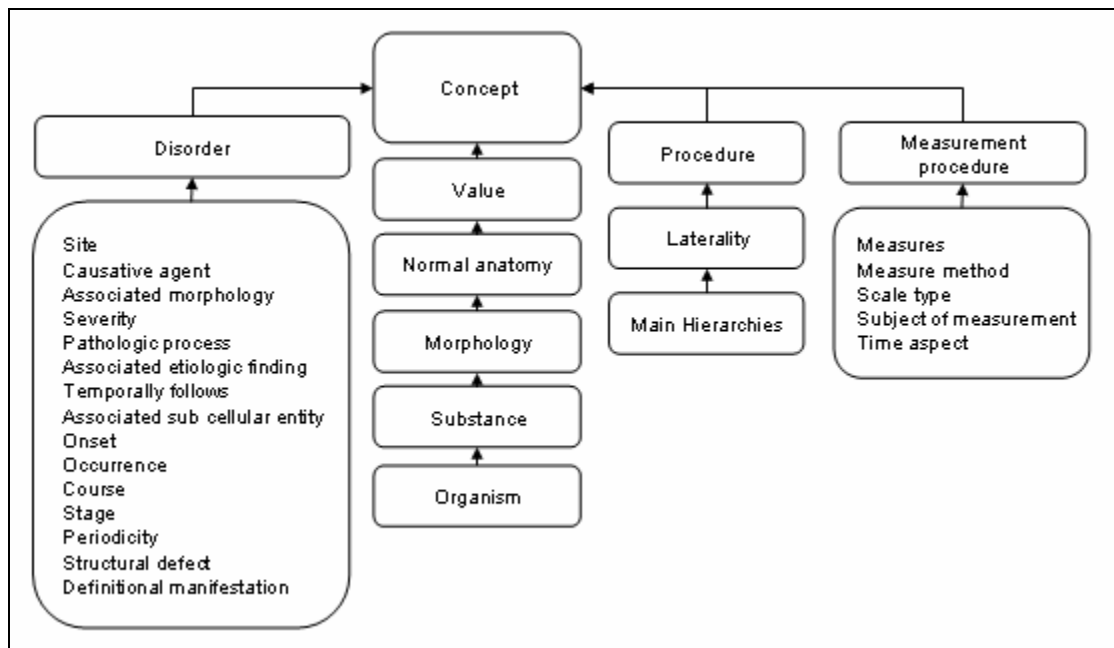
Historically, one part of the solution was the creation of the medical language; it was created in the light of medical facts for Latin and Greek languages. Moreover, today we find most these medical words still exist with slight morphologic changes. Similarly, we can see that the dynamic process of evolution of the medical language requires medical equivalent representations, and creates new relationships between medical terms and health accountability issues such as health billing and insurance policies.

In this way, one of the problematic topics of medical informatics is the handling of medical terminologies. Thereby, the challenge is twofold. First, the conflicting objectives in concept representation are similar to the real world. [Straub, 2002] Second, existing medical terminologies have shown us that current medical terminologies do not provide enough conceptual expressiveness to allow for an easy handling from a formal-logical point of view.

Health workers and health professionals can very easily recognize medical relationships. For instance, health workers, because of their training and experience, understand the relationships between ataxia and muscle function. These relationships are difficult to express for professionals without health training or experience. Furthermore, it is very complex when we deal with computers. Fortunately, mathematic foundations have good approaches such as probabilities, logic and Bayesian approaches. Given that, schemas like SNOMED relationship entities are very useful in the health entity relationship representation. Figure 2.1

Figure 2.1.1

Entity schema about relationships among medical concepts



The entity structure adapted from SNOMED<sup>®</sup> shows the relationships among disorders, concepts and medical procedures.

Figure 2.1.1 shows the entity schema that is used by SNOMED to represent complex relationships among medical concepts. According to this model, disorders have many features such as causative agent, severity, pathologic process, etc. and the main entity, named the concept, has relationships with the disorders, the procedure and the value which is the summary of relationships among organism, substance and morphologic concepts. This information suggests that we can represent complex entities in order to get effective standardization.

Given such extended standards, we could develop automated tools to support the task of updating mapped or merged terminologies. The work with patient-record systems, clinical decision-support applications, alerts and reminders, computer-based guidelines, and health care administrative systems has revealed that management of medical terminology and its evolution is crucial. However, these terminologies are not static because changes in health care are inevitable. Consequently, to share data and applications in health care, not only do we need standards for terminologies and concept representation, but we also for representing change.

In summary, the health industry has significant problems with information standardization because there are problems with medical terminology, medical information indexing and medical language processing. [Pacak et al. 1980]

### **2.1.2 Medical language**

Medical language is composed of medical terminology so as to assign meaning to different health processes. Nowadays, this qualitative appreciation (signs, symptoms and diagnosis) still classifies medical entities, but medical terminology has problems on structuring and indexing medical records. The main cause is the complexity of the medical language. [Zweigenbaum et al., 1999]

#### **2.1.2.1 The structural complexity of the medical language**

Medical language has an undesirable complex structure because it differs from more common natural language words. A medical word has several components and each of them carries meaning and contributes to the overall meaning of the word. [Thorsten et al., 2002]

Medical terms differ from current language words because they are frequently combinations of several Latin or Greek words. The Latin and Greek words shape around of 70% of medical words in languages such as English, Spanish and French. [Mario Nascimento et al., 1999] Each one of these words in Latin and Greek languages carries an important meaning and contributes to the learning of the medical term. For instance, a complex medical term is the name of a diagnosis “acropaquidermoperiostosis” which can be decomposed into its components:

## acro – paqui – dermo – peri – osto – osis

A simple word is the smallest unit in a sentence with meaning on its own. A meaningful word can be a simple word or a compound word comprising 2 or more simpler words – depending on the context. In most cases, the meaning of a compound word is more than just a combination of the meanings of the constituent simple words, i.e. some meaning is lost when the compound word is segmented in simple words. [Dai Y Lohn, 1999]

The same situation is given to any technical language; it can be a combination of stems, prefixes, suffixes and endings, and this is true for the medical language. In fact, it has valid components or components with meaning and invalid ones without important meaning.

The skeleton of a complex medical term has the following general schema. Figure 2

Figure 2.1.2.1

Structure of a complex medical term

Prefix	L	Stem 1	L	...	Stem n	Suffix	Euphony
--------	---	--------	---	-----	--------	--------	---------

Figure 2.1.2.1 illustrates the common structure of a complex medical term. According to this table; the grey cells of the structure have meaning such as the prefixes, stems, and suffixes; the white cells do not have significant meaning and are named the link-letters.

### 2.1.2.2 Medical word parts with meaning

The first part of a medical word generally has prefixes or stems for example, “a-“ stands for “without something”, “epi-“ stands for “over something”, “hemi-“ stands for “half something”, “poli-“ stands for “many or multiple things”, and stems like “neuro-“ that stands for words with neurologic reference or “cephalo-“ that stands for words with reference to the head. A stem always carries important meaning.

The suffixes are often endings that are used for plurals, verbs and attribute endings. However, in medical terms there are suffixes that carry further meaning, the suffix “-itis” stands for an inflammation and the suffix “-osis” stands for a degenerative process. They carry important meaning.

### 2.1.2.3 Medical word parts without meaning

These invalid components are the links-letters and euphonies. When stems are joined together in medical terms, they are usually connected with the links-letters such as “-e-“, “-i-“, “-o-“, “-eo-“, “-io-“, “-ico-“, or “-ato-“. The endings are generally named euphonies because they do not add

any meaning. They fix the final sound in the word. A stem can have different links and euphonies depending on the following stem, suffix, or euphonies. For example:

- “cardialgia”: “card” “-i-“ “algia”
- “cardiology”: “card” “-io-“ “logy”
- “cardiac” : “card” “-i-“ “ac”

The link-letters do not carry any meaning and thus can be discarded. Nevertheless, it is necessary to be careful because there are some stems with these letters as endings and for which these letters are only way to distinguish them from other stems; for example, “hemi-“ and “hemo-“ or “anti-“ and “ante-“. [Cole, 1995]

Although the components of such a medical word can also stand alone or in the place of an attribute, many medical concepts can be expressed in one term or with several words. These features carry several problems for parsers in medical language processing. Our work in this area has been to conceptualize these problems and to develop simple solutions described below.

### **2.1.3 Controlled medical vocabularies**

In the intent of solving these problems, the health industry has been using several controlled medical vocabularies such as: ICD [WHO, 1992], SNOMED [Roger et al., 1978], COSTART and WHOART [Saltzman, 1985], MEDRA [MedDRA, 2001] ICPC [Jamouille, 2002], MeSH [MeSH, 1997], UMLS [UMLS, 1997], DSM-IV [Frances et al., 1994], with the goal of mapping semantically equivalent terms such as *fever*, *pyrexia*, *hyperthermia*, and *febrile* with the same (numerical) value. Controlled vocabularies, an approximation on formal language, are very important when we need to disambiguate any vocabulary; nevertheless, many problems arise when we use codes in the medical field.

#### **2.1.3.1 Importance of controlled medical vocabularies in health care**

Computer-based systems that support health care require large controlled terminologies to manage names and meanings of data elements. According to Diane Oliver, “the controlled medical terminologies in computer-based patient-record systems facilitate data entry, data retrieval, data analysis, and data sharing.” [Oliver et al., 2000]

Controlled-medical-terminology systems share the requirements for a number of important tasks: [Bechhofer, 1994] searching for a term; [Bernauer, 1999] decoding and encoding; [Brachman et al., 1983] retrieval of information about concept meaning; [Brachman et al., 1985] translation to

other coding schemes; [Brachman et al., 1991] and management of additions, modifications, and obsolescence. However, since most existing systems have been developed independently and were not initially intended to interoperate, we have a plethora of differing controlled medical terminologies that use a wide variety of concept-representation techniques and methods for handling change.

When more than one terminology is essential for a task, terminology managers strive to handle diversity by mapping terminologies to one another with links between synonymous terms, or they incorporate one terminology in another to generate a combined, or merged, terminology. If medical vocabulary did not change, we could create mappings or generate merged terminologies. However, as medical knowledge, clinical practice and computer-application requirements change rapidly, change in terminology will always occur, even if they start with the same terminology.

#### 2.1.3.2 Medical concepts in controlled vocabularies

A review of terminology systems relevant potentially to medical care was discussed by Diane Oliver et al. They reviewed controlled terminologies such as the International Classification of Diseases, 9th Revision, Clinical Modification (ICD-9-CM), Medical Subject Headings (MeSH), the Diagnostic and Statistical Manual of Mental Disorders (DSM), the Systematized Nomenclature of Human and Veterinary Medicine (SNOMED), and the Unified Medical Language System (UMLS).

They also refer to other terminologies whose developers have published articles about modeling and maintenance, such as the James Read Codes [O'Neil, 1995; Robinson, 1997], the Medical Entities Dictionary (MED) [Cimino, 1994a; Cimino, 1994b; Cimino, 1996], and the General Architecture for Languages, Encyclopedias and Nomenclatures in Medicine (GALEN). Frame knowledge representation languages include KL-ONE [Brachman, 1985], CLASSIC [Resnick, 1993], LOOM [MacGregor, 1991], KRYPTON [Brachman, 1983], BACK [Peltason, 1991], Ontolingua [Gruber, 1993], and GRAIL [Horrocks, 1996; Rector, 1997b].

They found there is a wide variability in the approaches taken by medical terminologies and frame knowledge representation systems for concept representation. Figure 2.1.3.2 shows the comparison of these variations. As can be seen in this figure, controlled medical terminologies and frame knowledge representation languages have remarkable differences between constant unique codes and abbreviations.



Figure 2.1.3.2

Comparison of controlled vocabularies naming and identifying medical concepts

Feature	Controlled Medical Terminologies					Frame Knowledge-Representation Languages		
	I	Me	S	ME	U	C	GR	GF
Code	x	x	x	x	x			
Constant unique code			x	x	x			
Unique name	x	x	x	x	x	x	x	x
Changeable unique name	x	x	x	x	x	x		x
Synonyms		x	x	x	x	x	x	
Abbreviations					x			
Text definition		x		x	x			x
Translation to other coding schemes			x	x	x		x	
Multilingual translation		x	x		x		x	

*Legend:*  
 I: ICD-9-CM; Me: MeSH; S: SNOMED; ME: MED; U: UMLS;  
 C: CLASSIC; GR: GRAIL; GF: GFP.

Adapted from table 1 which was published by Diane Oliver in Representation of Change in Controlled Medical Terminologies. Stanford Medical Informatics, CA.

### 2.1.3.3 Relationships and hierarchies in controlled vocabularies

Oliver also showed the relevant features of concept organization, hierarchies, and other binary relations.

They found that controlled vocabularies are basically hierarchical, where concepts higher up in the hierarchy are more general than their descendants. However, the relationship between parent and child is not always named in controlled medical terminologies. In MeSH, for example, the relationship may be one whose implied meaning is IS-A, PART-OF, HAS-LOCATION, or CONTAINS. [Mc Cray et al., 1995]

In contrast, frame knowledge representation systems, in which automatic classification is of central importance, are rigorous about the relationship between parent and child: Every concept, except the top-level concept, must have at least one parent that subsumes it. A number of controlled medical terminologies have strict hierarchies, in which each concept can only have one parent. ICD-9, ICD-9-CM, SNOMED, and Read Version 2 use such hierarchies.

Figure 2.1.3.3

Comparison of controlled vocabularies by relationships and hierarchy

Feature	Controlled Medical Terminologies					Frame Knowledge-Representation Languages		
	I	Me	S	ME	U	C	GR	GF
Hierarchy	x	x	x	x	x <sup>1</sup>	x	x	x
IS-A hierarchy				x		x	x	x
Strict hierarchy	x		x					
Multiple parents		x		x	x <sup>1</sup>	x	x	x
Binary relations		x	x	x	x	x	x	x
Named binary relations				x	x	x	x	x
Binary-relation hierarchy							x	
Primitive and nonprimitive concepts				x		x	x	x
Transitivity of roles							x	
User-defined facets						x		x
Maximum cardinality						x		x
Minimum cardinality						x		x
Exact cardinality						x		x
Individuals						x	x	x
Inheritance				x		x	x	x

<sup>1</sup>The UMLS stores the hierarchies of its source vocabularies, but the hierarchies remain distinct.

*Legend:*  
I: ICD-9-CM; Me: MeSH; S: SNOMED; ME: MED; U: UMLS;  
C: CLASSIC; GR: GRAIL; GF: GFP.

Adapted from table 1 which was published by Diane Oliver in Representation of Change in Controlled Medical Terminologies. Stanford Medical Informatics, CA.

Figure 2.1.3.2 shows that the most common features between the controlled medical terminologies and the frame knowledge representation languages are the hierarchy, the binary relations, and the multiple parents. The least common features are the cardinalities and transitivity of roles.

Although the medical terminologies can represent similar concepts, recent studies have shown that the automatic translation among controlled medical vocabularies needs more work. It is the evidence that each vocabulary has a specific role in the health industry, according to the Table 2.1.3.2. There are poor results using the automated translation from SNOMED and Medlee to ICD-9.

Table 2.1.3.2

Inconsistencies in the automated translation between two terminologies

Automated Translation	Correct	Author
From SNOMED to ICD-9 – Automated mapping	40% Three-digit 30% Four-digit	[Franz et al., 1999]
From Medlee to ICD-9 – Automated encoding	69% codes	[Lussier et al., 2001]

## 2.1.4 The International Classification of Diseases - ICD

### 2.1.4.1 The ICD history

This classification originated as the “Bertillon Classification of Causes of Death,” prepared in the late 1800’s by Dr. Jacques Bertillon, chairman of the committee charged by the International Statistical Institute to prepare a classification of causes of death for international use.

The *Manual of the International Statistical Classification of disease, injuries, and causes of death* (ICD) has been enhanced for almost a century. It is widely used throughout the world to record causes of death and to classify diseases, injuries and related health problems. Currently, the World Health Organization (WHO) is in charge of its maintenance.

As a consequence, the WHO has been encouraging the use of the ICD, and its ninth revision (ICD-9) was adapted in the USA for clinical purposes in 1977. [WHO, 1969]

The 10th revision (ICD-10) was published in English in 1992. Translations of ICD-10 are now available in more than 20 languages. [WHO, 1992]

ICD-10 consists of three volumes, twenty chapters including six tabular lists of some 17,000 diseases and an alphabetical index for the diseases (13,005 entries in the Spanish version).

### 2.1.4.2 ICD role

The ICD has a crucial role in the process of sharing and comparing health data, specifically with morbidity and mortality statistics. Unfortunately, the health industry lacking standard vocabularies for other purposes such as accountability, billing and administration is adapting ICD terminology for purposes for which it was not intended. In one example, they believe that ICD can optimally support their needs. “Support the business needs of the organization (e.g. ICD9 or translatable to ICD9 for billing, insurance reimbursement, etc.)”. [Chin et al., 2001] Consequently, they have found many problems adapting and integrating ICD into their systems. [Krall et al., 1997]

In this paper, we propose that the purpose of the ICD is to promote international comparability in the collection, classification, analysis and presentation of mortality and mortality statistics. This role is sometimes forgotten and is consequently source of criticism. [Gibbons, 1999]

### 2.1.4.3 ICD-10

The tenth revision of the ICD was published in three volumes by WHO in the early 1990s. It incorporates the most fundamental changes to the ICD for almost 50 years and has been designed for use well into the 21<sup>st</sup> century. Despite WHO’s recommended implementation date of 1993, its

introduction for mortality coding has generally been relatively slow, though a few countries did start to use it in the mid-1990s. The most obvious change is the move from a 3 or 4-digit numeric code to 3 or 4-character alphanumeric code. For example, malignant neoplasm of stomach has changed from **151** to **C16** in ICD-10 and a fourth digit may be used for sub-categories. The introduction of an alphabetic first character greatly increases the potential number of codes available. In fact, ICD-10 uses a total of some 8,000 unique codes, over 3,000 more than ICD-9. [Anderson et al., 2001]

In summary, medical language differs from non-medical natural language because it is very agglutinated. Medical language has a crucial role in the health industry, but it has some problems associated to the issues of the best health knowledge representation because of the complex association among meaning, sign, and symbol. In fact, the most complex part is the relationships among medical words because they are difficult to express for non-related health professionals. With the intent of solving these problems, the health industry has been using several controlled medical vocabularies as ICD which has a crucial role in the process of sharing and comparing health data, specifically with mortality statistics. Unfortunately, the health industry is using this vocabulary for other purposes such as accountability or billing.

## **2.2 Problems with data entry and human errors**

The recording and indexing process of health data is more problematic than the medical language because it integrates in the “human factor”. Accordingly, in this part I review relevant research about data entry issues and coding human errors, and how they occur.

### **2.2.1 Data entry**

#### 2.2.1.1 Goal of data entry and coding

The goal of data entry and coding is the best “hit rate”, a proportion of code terms that preserves accuracy and consistency and expedites handling of uncoded terms. Indeed, Weber classified the data entry problem in three components: stability (the ability of a coder to consistently assign the same code to a given text), reproducibility (intercoder reliability) and accuracy (the ability of a group of coders to conform to a standard). [Weber, 1990]

#### 2.2.1.2 Components of a coding system

An automated coding process has several components. The first element is normally an input such as dictionaries and verbatim terms. This process uses lexical transformations to normalize

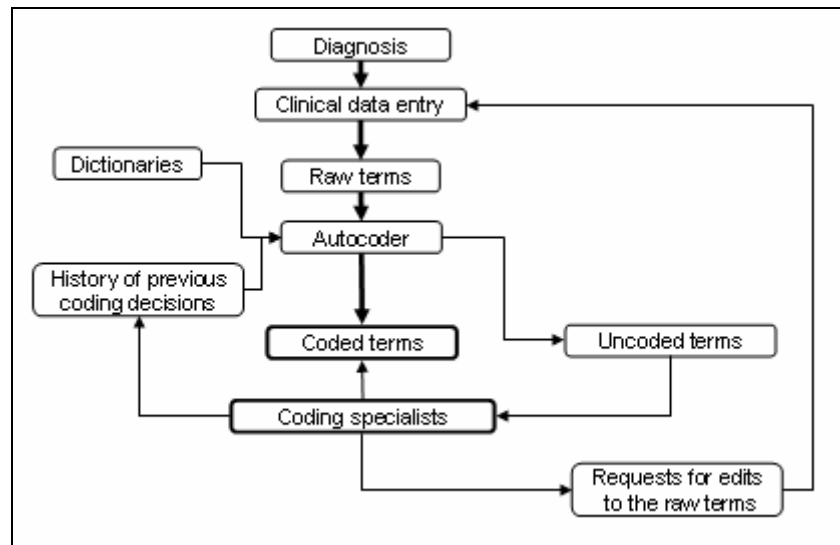
text, and lookup steps that attempt to match inputs. Another element is the outputs, which are usually the coded terms and uncoded terms. [Jablin, 1992]

### 2.2.1.3 The coding workflow

All of these elements are shown in figure 2.2.1, which represents the coding workflow in the health industry. As we can see, the workflow begins with the clinical data entry (in this case a diagnosis), which is transformed in coded or uncoded terms by the Autocoder using dictionaries and coding decisions and when there are problems. In complex cases the coding specialist resolves the final coding.

Figure 2.2.1

Health coding workflow for diagnosis



Schema of coding workflow for diagnosis adapted from “Clinical trial Classify for Automated Medical Coding” which was published by Jodi Greenspan Project Director - Belmont Research Inc.

### 2.2.1.4 The problem with coding data

Coding health data suffers from poor acceptance, in particular because it is often badly perceived by health workers. [Brett, 1998] Moreover, mandatory coding only used for financial purposes may result in codes of poor quality, as observed for Medicare patients in US hospitals. [Hsia et al., 1988]

### 2.2.1.5 One of the solutions – human coders

Employing human coders trained in both health rules and coding rules solve one part of the problem. However, the problem of coding health data can go beyond the vocabulary. Because of the characteristics of the medical domain, many techniques that work well for less demanding

data sets are not practical for autocoding, and properly leveraging the knowledge of human coders remains the key to developing and deploying successful systems. [Kornai et al., 2001]

#### 2.2.1.6 Other problems with coding data

The process of data entry has more troublesome issues such as workflow, coding acceptability and the coder's medical knowledge. Not only is it complex, but also the coder, a human, adds other human problems. Indeed, the data entry process deals with the text inconsistencies, and copes with the problems of: [Rowe et al., 1990]

1. Rearranged words
2. Missing words
3. Spelling variations
4. Abbreviations
5. Various punctuations.
6. Plural vs. singular forms
7. Extraneous words
8. Synonyms
9. Inconsistent hyphenation
10. Syntax

Some of these problems depend on the coder's typing skills, job concentration, and computing experience. In reality, there are problems that depend completely on human nature; these are named human coding errors.

### 2.2.2 Human errors

#### 2.2.2.1 Manual coding

The process of manual coding is a large problem because when human beings use characters, numbers or letters, we can recognize five kinds of human errors in data entry. [Verhoeff, 1969; Wagner, 1989; Ali, 1992]

According to Richard Hamming [Hamming, 1986], by far the two most typical human errors are:

- Interchanging adjacent digits: *ab* becomes *ba*
- Doubling the wrong one of a triple of digits, two adjacent ones of which are the same: *aab* becomes *abb*

J. Verhoeff gives a more detailed categorization of the sorts of errors humans make in dealing with decimal numbers, based on a study of 12000 errors: [Wagner et al., 1989]

- Single errors: *a* becomes *b* (60% to 95% of all errors)

- Omitting or adding a digit (10% to 20%)
- Adjacent transpositions: *ab* becomes *ba* (10% to 20%)
- Twin errors: *aa* becomes *bb* (0.5% to 1.5%)
- Jump transpositions: *acb* becomes *bca* (0.5% to 1.5%)
- Jump twin errors: *aca* becomes *bcb* (below 1%) [Lower for longer jumps]
- Phonetic errors: *a0* becomes *1a* [since the two have similar pronunciations in some languages. (0.5% to 1.5%)

In the explanations above, *a* is not equal to *b*, but *c* can be any decimal digit.

We can easily eliminate or detect the problem of omitting or adding digits by restricting the input field to a given number of digits if we deal with numbers which are fixed in format, such as Social Security Numbers or other ID numbers.

Other errors are detected by calculating whether the check equation for a particular check digit or the whole scheme. If the equation is not true, an error is present; if it is true, there may or may not be an error.

#### 2.2.2.2 Suggested schemas for detecting human errors

A number of different schemes for detecting decimal number errors have been suggested, and several are in particular use. In the following list, four schemes are outlined.

1. The International Standard Book Number (ISBN) uses a weighted code. [Rosen et al., 1990]
2. The "IBM check", is an even/odd weighted code. [Wagner et al., 1989]
3. The Universal Product Code (UPC) uses a weighting factor of 3 overlooking adjacent transpositions of digits. [Goodaire et al., 1998]
4. Verhoeff proposed a scheme based on multiplication in the dihedral group  $D_5$ , which is not commutative ( $a*b$  is not always equal to  $b*a$ ).

Verhoeff's check equation catches all single errors, all adjacent transpositions, over 95% of twin errors, over 94% of jump transpositions and jump twin errors, and lots of phonetic errors. [SNOMED, 2002]

### 2.2.3 Errors coding health data

#### 2.2.3.1 Errors coding mortality data

Before the use of computers, the coding process was essentially manual and the use of health workers trained in coding registers, certificates and records is now well-known. Given that, multiple studies developed in different countries have found that manual coding is poor. The rates

of human error have an range of 2% to 20% according to the International Collaboration effort on automating mortality statistics. They report multiple problems such as wrong codes, missed codes and errors in underlying cause of death.

Another study of the Canadian Bureau Statistics demonstrated that the error rates with manual coding (coding health data without computers) were largely attributed to spelling errors, multiple responses, abbreviations, and missing entries in the reference files. [Miller et al., 1991] However, there are also errors in automated coding (coding health data with the support a compute) from free text to numeric codes when the code is very complex. According to the figure 2.2.3.1, there is a range of reliability and accuracy coding health data to controlled vocabularies. The most successful studies used techniques such as automated coding resulting in 98.2% of correctness, and natural language processing with 98%.

Table 2.2.3.1

Errors coding from health data to health codes

<b>Source from – to / Process</b>	<b>Correct</b>	<b>Author</b>
Free text - ICPC / Automated matching	92% Chapter, 80% Code level	[Letrilliart et al., 2001]
Diagnoses - ICD / Automated matching	79% codes	[Rada, 1990]
Cause of death - ICD / Automated coding	81% codes	[WHO, 1998]
Diagnoses - ICD / Automated coding	98.2% codes	[Blanquet et al., 1990]
Automated coding using text recognition	90% unique 72% multiple	[Macchia et al., 1999]
Medlee - ICD / Automated encoding	69% codes	[Lussier et al., 1999]
Diagnosis - ICD / Natural language processing	98% codes	[Lovis et al., 1998]

ICPC: International Classification of Primary Care

ICD: International Classification of Diseases

Medlee: Natural Language processing system

### 2.2.3.2 Errors in pragmatic relationships

Related concepts have exclusive relationships between a term and a hierarchical family. This exclusivity has levels and conditions and that is one of the frequent problems with medical vocabularies today. In our research, these concerns are very important since health coders often do not understand medical or health relationships. For instance, words such as “tuberculosis” and “cholera” have no a pragmatic relationship. In this sense, there are clusters of relationships among the medical words that are sometimes known by health professionals. Consequently, it is one of the causes of poor quality in health data.

### 2.2.3.3 Errors classifying cause of death

Until recently, the underlying causes of death have been routinely coded and analyzed in national mortality statistics. This means that other diagnosis information on death certificates, regarded by certifying clinicians as relevant to the death, have been discarded. Furthermore, the rules and



methods for selecting the underlying cause of death have changed with the introduction of ICD-10 coding rules. For instance, myocardial infarction (MI) as the primary cause of death has shown a large decrease in mortality rates in recent decades because WHO does not allow MI as an underlying cause of death. [Goldacre et al., 2003]

Traditionally, tabulations of mortality statistics have presented information based on a single cause for each death and the early international classifications were devised to categorize the single cause normally reported on death certificates. However, as doctors began to report more than one condition on certificates, it became necessary to develop rules to select a principal or 'underlying' cause.

In the ICD the underlying cause is defined as

- The disease or injury which initiated the train of morbid events leading directly to death or
- The circumstance of the accident or violence that produced the fatal injury

Speaking specifically about these issues, there are rules for coding mortality data. The use of ICD-10 incorporates important definitions. For example, the following key definitions are described in the Volumes I, II and III of the ICD-10 Spanish version.

List of adopted definitions for the WHO, numbers WHA20.19 and WHA43.24 – Article 23 of the WHO:

- Causes of death: are the causes of death that have to be registered in the medical death certificate and also are the whole disease, illness and lesion that cause or contribute to the death. They include the accidental or violence circumstances that produce these lesions.
- Basic causes of death: is the underlying cause of death that can be diseases and lesions that begin a pathologic event and will finish in death or accidental or violent circumstances that produce death.

Currently, the WHO has published the coding rules; however, they are continually working on problems with these rules. Specifically, there is a specialized group that reviews suggestions and convenes meetings in order to achieve good coding rules for ICD-10.

## **2.2.4 Coding health software**

### 2.2.4.1 Coding principles

In order to enable accurate coding we need to understand how a coding system works and how it codes accurately. The process of coding data is based on principles in which a computer-coding tool is correct. Indeed, a system assigned code is considered correct when it agrees with a "truth"

code assigned by coding experts for the case in question. Which "truth" code to assign to a case is not always apparent, and such a disagreement between the computer assigned code and the "truth" code does not always represent an error. Cases for which the computer assigns an incorrect code are named mismatches. Each automated coding system calculates a score for each code that it produces. Reliability also depends on the code produced. Computer-coding systems could improve this in some cases. In order to control errors, a minimum acceptable score, or cutoff score, is determined for each code category. A computer-assigned code is a score in which the cutoff is above the code category. [Rowe et al., 1994]

In 1990, the Statistical Research Division (SRD) of the Census Bureau for performing automated coding conducted research on improved methods. This research has two areas where computer-coding systems are useful: fully automated coding and computer-assisted clerical coding. The fully automated systems are used for large batch operations and the computer-assisted clerical systems are used for residual coding of cases that a fully automated system could not resolve. This research program also focused on understanding algorithm characteristics, improving its productivity, reducing errors, and applying it in new situations. [Gillman et al., 1990]

#### 2.2.4.2 Automated coding of ICD-10

Automated coding in medical field is the process of assigning automatically or interactively a standard code to reported text (verbatim) for medical terminologies using controlled industry-standard dictionaries such as ICD, CM, CPT, WHODRL, etc. [Appel et al., 1983]

Although the ICD-10 is used in forty-seven countries for coding national mortality data, only twenty-five countries use software produced by NCHS. Currently, this DOS based solution is not able to use the advantages of ICD-10 over ICD-9 codes. Because ICD-10 has more detailed data, higher quality and more timely data than ICD-9, as well as the routine production of multiple cause of mortality ideally automated systems should use ICD-10. [CDC-NCHS, 1999]

As a consequence, the integration of computer-assisted tools for coding of ICD-10 for mortality data must be accurate, scalable, and adequate for each country. More specifically, the group named the International Collaborative Effort on Automating Mortality Statistics (ICE) [CDC-NCHS, 1996] has reported many problems among member countries owing to several differences of nomenclature, decision tables, health workflow of vital records, coding process, language and interpretation of ICD-9 rules. [Arialdi et al., 2001]

In similar efforts, the Centers of Disease Control of the National Center for Health Statistics (NCHS) and ICE are working together with the purpose of sharing knowledge and experience with automated systems for coding mortality information, to develop and improve existing automated systems through collaboration, and to facilitate the transition to the tenth revision ICD-10 [WHO, 1992]. This effort of NCHS began in 1970's when they developed the first program software for coding the underlying cause of death, named ACME, which stands for the "Automated Classification of Medical Entities" [NCHS, 1983]. Since then, the NCHS has added three additional inter-related modules for processing mortality data such as TRANSAX, MICAR, and SuperMICAR, which are coding software that retrieve and classify mortality data in ICD-9 codes in the USA. [Chamblee et al., 1978]

The use of these software programs in Europe began in 1990s, but there were errors in the process of integration. For instance, in 1993 Sweden showed 7.2% of errors before the integration of MIKADO (a Sweden coding system) [Johansson, 2001] in 1994 Spain showed 20% of errors before the use of ACME, and UK showed 20% of errors with TRACER. [Cleone, 2001]

In view of the limitations of ICD-9-CM, in June 1998, the Health Care Financing Administration (HCFA) responsible for the maintenance of the procedure coding system for reporting inpatient procedures for Medicare and Medicaid, contracted with 3M Health Information Systems to develop a new procedure coding system to be used with the forthcoming disease coding system, the International Classification of Diseases, 10<sup>th</sup> Revision, Clinical Modification (ICD- 10-CM), being developed by the United States National Center for Health Statistics. Therefore, the current version of ICD-10-PCS is reminiscent of SNOMED-6, which was originated as a modular system in which every diagnosis and procedure was constructed from components. [Slee, 1998]

Currently, Peru is undergoing the transition from ICD-9 to ICD-10, but no one software exists for coding ICD-10. As a consequence, the delay in reporting vital health statistics in Peru is a very big problem and as a result of having studied all these concerns, I have performed an analysis of the mortality data in Peru.

In summary, coding health data has numerous problems such as coding acceptability, and coder's medical knowledge. It is not only complex, but also the health coder, a human, adds rearranged words, misspelled words, and inconsistent syntax. This is due to the fact that humans make errors typing numbers or characters. Recent studies have reported that coding errors were largely attributed to spelling errors and missing entries, for instance. Diverse organizations from

developed countries, worried about coding errors in health data, have been using coding software program with relative success, but in other countries, including developing countries, there are still problems with coding errors.

## **2.3 Problems with vital statistics in Peru**

### **2.3.1 Vital statistics in Peru**

#### 2.3.1.1 Workflow of the death certificates in Peru

Vital statistics of Peru are obtained from the official records of live births, deaths, fetal deaths, marriages, divorces, and annulments. The official recording of these events is the responsibility of the individual departments and independent registration areas in which the event occurs, and the Government obtains use of the records for statistical purposes through a cooperative arrangement with the responsible agency in each department.

In recent years, the attention has been focused on improving the quality of vital statistics and making them useful and available. The interest in vital statistics widened when Governmental agencies were challenged to define needs for and effects of various national health and welfare programs, and they began looking for pertinent and reliable statistics on which to base decisions. [WHO, 1997]

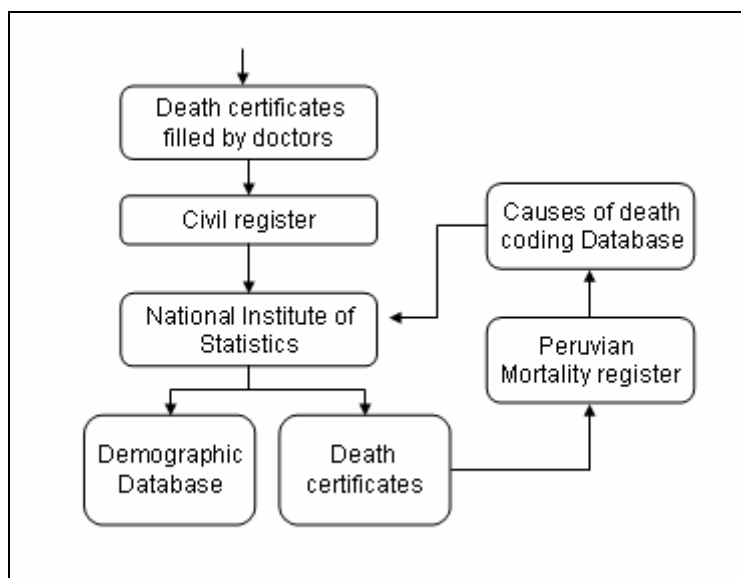
The registration certificates assumed new importance as they were looked to as a source of credible national vital and health statistics for use by all levels of government, institutions, and the general public. As health and social problems became more complex, the content of the information collected on the vital records was expanded and measures to improve its quality and usefulness were added because vitals statistics are vital indicators for the country. [INEI, 1995]

One of these vital indicators is mortality collected using death certificates. The death certificate is a legal document that provides the basic cause of death classified for purposes of statistical tabulations. This document and the rules that define how it is filled out place responsibility on the physician for reporting causes of death in such an order that the underlying cause of death is indicated because of its position on the certification form. However, sometimes, there are not adequate mechanisms to distinguish between properly and improperly reported cause of death. [NCHS-CDC, 1997]

Although there are models of coding workflow, the coding death certificates workflow in Peru do not incorporate information systems in its structure, according to the following figure.

Figure 2.3.1.1

The death certificates workflow in Peru.



Death certificates workflow made by the author following vital statistical guidelines from Peru.

Figure 2.3.1.1 shows the Peruvian coding workflow around death certificates. As we can see from this figure, the death certificates, completed by doctors, are registered in Peruvian civil register offices and then recorded in the National Institute of Statistics that produces the Peruvian demographic database. This Institute also collects the death certificates and generates the Peruvian mortality register. [INEI, 1995]

### 2.3.1.2 Errors in mortality data using the ICD-10 in Peru

When we code mortality data with sex and age, we can often create inconsistent data, because there are groups of causes of death that have restrictions by sex and age such as gynecologic diseases or neonatal cancers. Also, there are groups of causes of death without sex or age restrictions such as trauma and neurologic diseases. In this sense, part of the solution to the problem is based on the effective use of the following list of causes of death:

- a. Causes of death without sex or age restrictions
- b. Causes of death with sex restrictions and without age restrictions
- c. Causes of death with age restrictions and without sex restrictions
- d. Causes of death with sex and age restrictions

My initial work was to recognize the level of error with sex, age and ICD-10 codes in Peru. With these concerns in mind, I designed a descriptive analysis of coding errors in Peru using the Peruvian mortality data from the Peruvian Ministry of Health. This work was developed using mortality data from 1999 to 2000. The preliminary findings showed a poor quality of data due to several different kind of errors, in some cases made by physicians (choosing the basic cause of death) and other cases by health coders (mistyping the ICD-10 code, age and sex). Table 2.3.1.2 shows a summary of these coding errors in Peru. According to this information, we can see that the most common coding error was the coding error for external cause of death at 92%, and the lowest coding error was the error in the value of the ICD-10. External cause of death means an accident or violent trauma that leads to the death.

Table 2.3.1.2

Description of coding errors using ICD-10 - Ministry of Health from Peru

Variables	1999*		2000**	
	Frequency	%	Frequency	%
Error in value ICD-10	198	0.23		
Error in underlying cause of death	15,836	18.29	14,872	17.68
Error of underlying cause of death type *	190	0.22	22	0.03
Error of underlying cause of death type N	16,027	18.51	14,872	17.68
Error of sex with ICD-10	2	0.00	9	0.01
Error of sex with underlying cause of death	16,217	18.73	14,894	17.71
Error frequency	4	0.00	3	0.00
Error of probability	4,812	5.56	3,248	3.86
Error in minimum age for cause of death	153	0.18	94	0.11
Error in age with ICD-10	12,682	14.65	12,309	14.64
Error in ZIP codes	13,786	15.92	13,505	16.06
Error in value of extern cause of death	79,550	91.88	75,824	90.15
Error in value ICD-10 extern cause of death	578	0.67	16	0.02
Empty values in extern cause of death	78,972	91.22	75,808	90.13

\*86,659, \*\*84,105 cases.

Source of data: Mortality data of 1999 and 2000. Health Ministry from Peru

Almost eighteen percent of the data had errors between sex and cause of death. For instance, the data included men with gynecologic causes of death. Fifteen percent of the data had errors between the range of age and the cause of death. In this case, I found children with causes of death only occurring in elderly people.

We also have found errors for ZIP codes, place of death, and kind of health worker who registered the death.

In conclusion, these coding errors findings give us the necessary background to develop tables of decision for coding ICD-10 because our validation of the data in the mortality analysis was made using tables of decision.

### **Summary**

In this chapter I have given a review of the relevant literature about parsing methods, coding health data, medical language complexity, controlled vocabularies and problems with human errors in coding.

Standardization of medical knowledge into numeric representation represents a difficult challenge for medical language processing. Coupled with the maintenance of multiple classifications necessitates identifying consistent hierarchical placement. Historically, one part of the solution has been the creation of medical language and this has led to one of the problematic topics in medical informatics the handling of medical terminologies. In fact, medical terminology indeed draws heavily on structuring and indexing medical records; the main cause is the complexity of the medical language.

With the intent of solving these issues, the medical field has been using several controlled medical vocabularies as ICD and more vocabularies, with the goal of mapping semantically medical equivalent terms.

Although ICD-10 has been developed and enhanced for more than a century, it is not widely used throughout the world to record causes of death and to classify diseases, injuries and related health problems. This is despite the fact that it incorporates the most fundamental changes to the ICD for almost 50 years and has been designed for use well into the 21<sup>st</sup> century.

Coding health data is, however, meeting with poor acceptability, in particular because it is often perceived of no direct clinical value for the physician. Moreover, mandatory coding only used for financial purposes may result in codes of poor quality. Hence, the process of data entry has additional problems such as workflow, coding acceptability and coder's medical knowledge. Not only is it complex, but also the coder, a human, adds other problems because when human beings use characters, numbers or letters, there are many kinds of human errors from mistyping such as interchanging adjacent digits, omitting or adding ones.

Although the ICD-10 is used in forty-seven countries for coding national mortality data, only twenty-five countries are using DOS based ICD-9 coding software. Currently, Peru is going through the transition from ICD-9 to ICD-10, but no one software exists for coding ICD-10. As a consequence, the delay in reporting vital health statistics in Peru is a very large problem.

Therefore, having studied all these concerns, our analysis of the mortality data in Peru showed very poor quality.



## CHAPTER 3

### 3. The research approach and Framework

This chapter describes the aims and the framework that is used for the purposes of my research. A simple three-step modeling methodology that has been successfully used to develop IcodeX is described. The methodology section focuses on the coding rules approach and discusses the role of the main steps coding mortality data. The specific explanation about the methods and how it was performed is shown in the appendix B.

#### 3.1 Aims

In the light of our facts, my principal aim is to develop a computer assisted tool for accurate coding of ICD-10 for mortality data (IcodeX). In this sense, my specific aims are:

1. To determine the problems behind the manual coding for mortality data
2. To determine the level of coding errors in Peru
3. To develop an mapping algorithm for coding of ICD-10 for mortality data
4. To test the developed application

#### 3.2 Methodology

The first part of my work was focused on determining the problems behind the manual coding for mortality data and the analysis of coding errors of mortality data in Peru.

The work in this phase, shown in chapter two, has two parts: a bibliographic review of relevant research about problems (data entry errors and medical vocabulary) and an accurate analysis of errors for mortality data in Peru.

The mortality analysis was performed following these three specific tasks:

1. Collection of mortality data
2. Discovery of errors using ICD-10 lists
3. Descriptive analysis

I have used mortality data from the Ministry of Health from Peru from 1999 to 2000 to find errors using ICD-10 in Peru. The process of error recognition was performed using extensive lists of validation and ICD-10 coding rules. In this sense, I was using the coding rules published in the volumes I, II and III of the ICD-10 books. After the reliability analysis, I performed the descriptive analysis using statistical software.

##### 3.2.1 Intended basic functions for an accurate coding

With the coding error findings in mind, the following development of the tool's algorithms was organized in the following basic functions for an accurate coding.

- Control of inputs: How well the inputs such as sex, age and cause of death are managed without errors.
- Control of sequence of errors: How well an error is immediately controlled avoiding more errors
- Filter of special characters: How well works with accentuated vowels and special characters without such as %, &, \*, #, @ etc.
- Control of pragmatic relationships among medical words
- Control of coding rules for underlying cause of death
- Control of coding rules for sex and underlying cause of death
- Control of coding rules for ranges of age and underlying cause of death

### 3.2.2 Planned main tasks

Although there are plenty of ways to solve the coding errors, I propose a new strategy, in order to follow the mortality coding rules from Peru, designed in two main tasks. The first task is the control of errors in cause of death controlled by sex and age as well as the indirect help for the second task, the search for ICD codes. The following figure 3.2.2-A shows us how algorithm would deal with mortality data and ICD rules.

Figure 3.2.2-A

Control of errors in cause of death controlled by sex and age

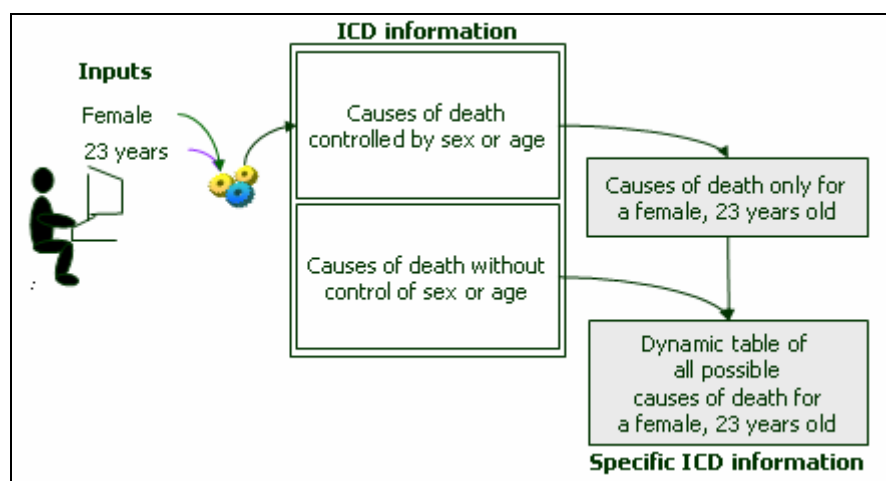


Figure 3.2.2-A shows our strategy in the control of input errors between inputs (sex and age) and the cause of death. According to this figure, the whole list of underlying causes of death is separated in two parts. The first part is the causes of death with control of sex and age and the second one does not have this control. Then, the first part could be separated in small tables using the ICD coding rules. This process permits us to control future problems with these problems. For instance, for a deceased female of 23 years old, the process will select a specific table with the whole list of possible causes of death for a female of 23 years old such as gynecologic problems or trauma, but not testicular cancer or neonatal disorders.

The second task works with the cause of death; basically, it auto completes the rest of the index word and expands the ICD-10 index of diseases so as to get the correct ICD-10 code. The following figure 3.2.2-B represents the second part using the small table from earlier figure.

Figure 3.2.2-B

Working with the cause of death

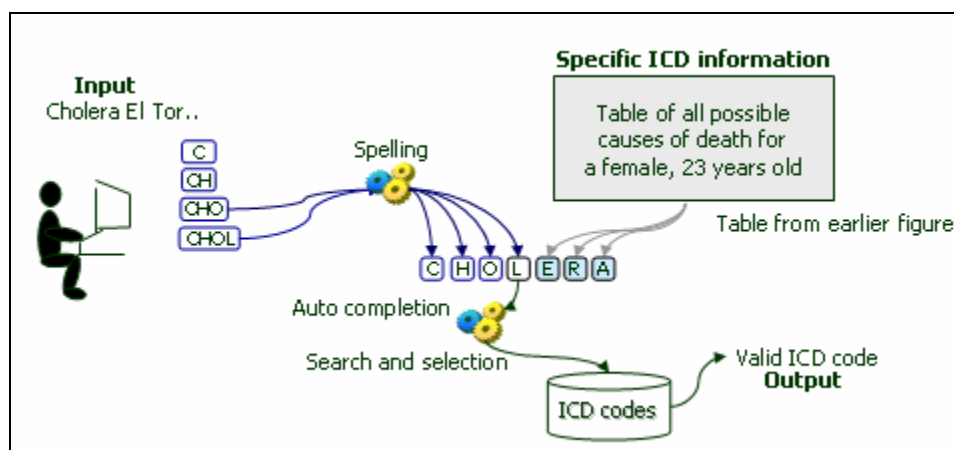


Figure 3.2.2-B shows the work with the cause of death. Having selected the specific table with all the possible causes of death for a combination of sex and age, the user is ready to enter the cause of death. In this sense, the system saves time for future errors that can be done by the coder.

The new event using the cause of death is basically performed using new functions such as auto completion and selection of the ICD code; moreover, the system increases its speed and saves time because the pre selected table is small.

Finally, the IcodeX evaluation allows us to discover the maximum performance for IcodeX or to uncover anomalies or instabilities that may only become apparent over an extended time period or under high usage.

For this phase, the aim is to find out what is and is not coding well on the application and my specific questions to ask for end-users are the following questions:

- Can IcodeX code successfully?
- How fast does IcodeX code?
- How many errors there are in the coding process with our tool?
- Are there errors about incorrect coding?
- Are there errors about impossible mortality data?

### **Summary**

This chapter showing my strategies is representing the main coding contexts. In fact, I have designed two events. The first works with the inputs of age and sex and the second with the underlying cause of death. Both events were used in the development of the coding tool.

## CHAPTER 4

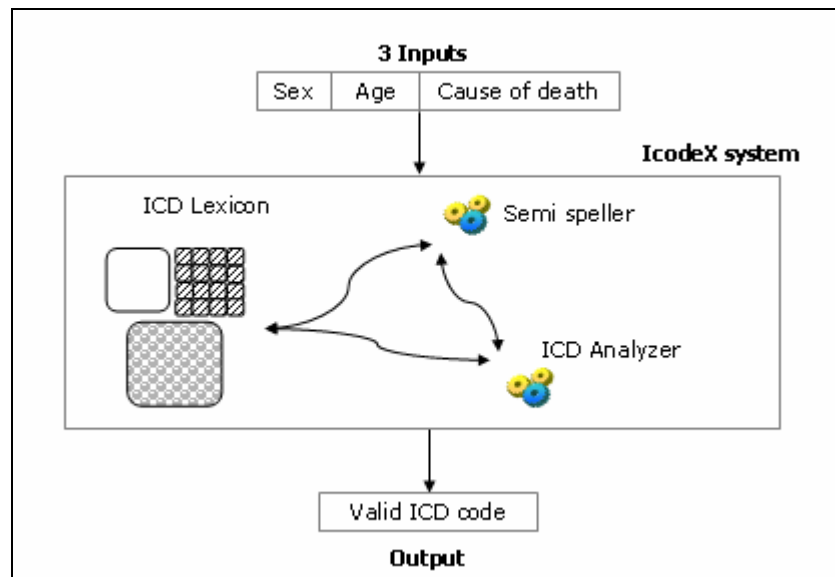
### 4. IcodeX system description

Chapter 4 provides details of IcodeX. The IcodeX system is composed of three parts and this chapter demonstrates the modular algorithms of IcodeX such as the semi-speller and the ICD analyzer, plus the semantic and syntactic checks required to detect input errors encountered in coding mortality data. As will be demonstrated, the main algorithms organized in order to achieve improved performance advantages, including a combination of top-down parsing techniques showing speed and efficiency.

As we can see in the figure 4, three inputs (sex, age and cause of death) are required by the IcodeX system to get a valid ICD code, the output.

Figure 4

The IcodeX system description



#### 4.1 The ICD Lexicon

The Lexicon is an important part of an integrated coding tool. Using the ICD-10 information, I made the lexicon of IcodeX (ICDL). Basically, I only rearranged the ICD-10 structure respecting the sensitive ICD-10 information using the coding rules from the ICD-10, which were published by WHO in 1990. In fact, the ICDL includes the entire ICD-10 information set. I set up the ICDL organization and structure to facilitate efficient access time, spelling correction, and accurate coding.

##### 4.1.1 The ICDL structure

The design of the ICDL structure is premised on the following objectives:

- Fast access to ICDL: The time spent to extract the ICD information for an existing cause of death in the ICDL should be minimal. In other words, the structure must support a fast access method.
- The text of causes of death must be available to the spelling correction without any restrictions. The text must be alphabetically sorted to allow the spelling corrector to perform its task.
- The index of diseases as well as the whole description must be easily accessible to the ICD analyzer.
- The structure should be specific so as to avoid redundant data

As discussed in chapter 2, the ICD information includes the cause of death and its category (referred to as the index of categories [IC]), in order to follow the hierarchical ICD four-digit disease code structure as well as their categories, which have three-digit codes. According to the hierarchical structure of the ICD-10, the cause-of-death code is the combination of the category code and an identifier code. The codes involved with the category “Cholera” are shown in table 4.1.1

Table 4.1.1

Three-digit and four-digit codes of the ICD structure

Three-digit ICD-10	Four-digit ICD-10	Complete description
A00	A00.0	Cholera caused by <i>Vibrio cholerae</i> 01, biotype cholerae
A00	A00.1	Cholera caused by <i>Vibrio cholerae</i> 01, biotype El Tor
A00	A00.9	Cholera no specified

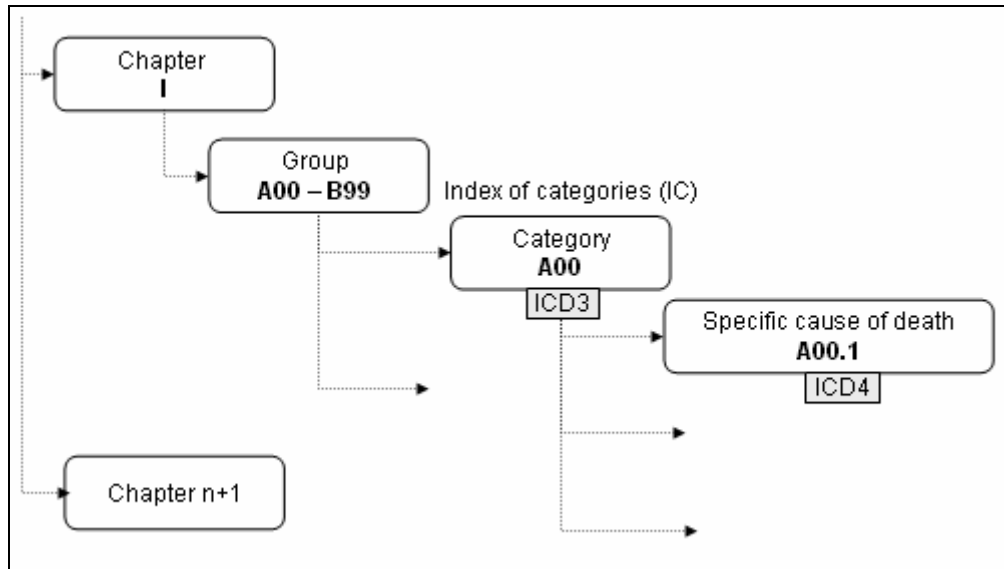
We are using the three-digit IC codes contract the size of the ICD information since the categories from the IC are generally a single word (7%), two words (10%) and three words (15%); however, for categories with more than four words the first word is more important than the other are.

Figure 4.1 demonstrates a hierarchical composition in the ICD-10 structure; each chapter contains groups, each group contains categories of cause of death, and each category contains a group of category-related causes of death. The two last parts of the ICD structure, index of categories and specific cause of death, have codes. In fact, future references to these codes will include their acronyms for the specific cause of death as ICD4 and for its category’s code as ICD3. For example, consider the hierarchical structure for the specific cause of death A00.1. According to

the figure 4.1, its category is A00, its group is A00-B99 (the group of infections diseases), and its chapter is the chapter I.

Figure 4.1

Hierarchical structure of the A00.1 code – Cholera caused by El Tor



#### 4.1.2 The decision matrix table

The reorganized ICD-10 information, the ICDL, is accessed by the ICD analyzer using a decision matrix table that I made using the coding mortality rule information from the ICD-10 information. The ICDL structure consists of a collection of tables organized in a decision matrix table (DMT), which is then analyzed together with the inputs of age and sex by the ICD analyzer to get the correct table.

As discussed earlier, coding rules are used to obtain an accurate ICD code. Therefore, there are groups of codes determined by sex, age, or the interaction of both variables. In addition there are causes of death determined without consideration of the interaction of age or sex. With this in mind, I separated the ICDL structure into twenty-one smaller tables, organized by sex and age.

In the construction of each table I used five variables including the sex, the minimum allowed age, the unit of the minimum allowed age, the maximum allowed age and the unit of the maximum allowed age. In this algorithm, each table includes all the possible underlying causes of death for a specific combination of age and sex. Table 4.1.2 shows the composition of the DMT structure.

Table 4.1.2

Structure of the decision matrix table

Sex	Age from	Age to	Age unit	Table number
Any	Any	Any	Any	22
Female	0	0	Hours	1
Male	0	0	Hours	12
Female	1	24	Hours	5
Male	1	24	Hours	15
Female	2	6	Days	8
Female	7	28	Days	11
Male	2	6	Days	18
Male	7	28	Days	20
Female	1	48	Weeks	6
Male	1	48	Weeks	16
Female	1	1	Years	4
Male	1	1	Years	14
Female	2	4	Years	7
Male	2	4	Years	17
Female	5	11	Years	10
Male	5	11	Years	19
Female	12	14	Years	2
Male	12	14	Years	13
Female	15	49	Years	3
Female	50	99	Years	9
Male	15	99	Years	21

As can be seen in table 4.1.2, the table number 22 is the table that contains the ICD codes without restrictions for any sex or age. This table is always loaded by default before the ICD analyzer can load any table from 1 to 20 according to the combination of sex / age.

#### 4.1.3 Improving methods of access to the ICDL

These described arrangements greatly permit the ICD analyzer to control these three problems:

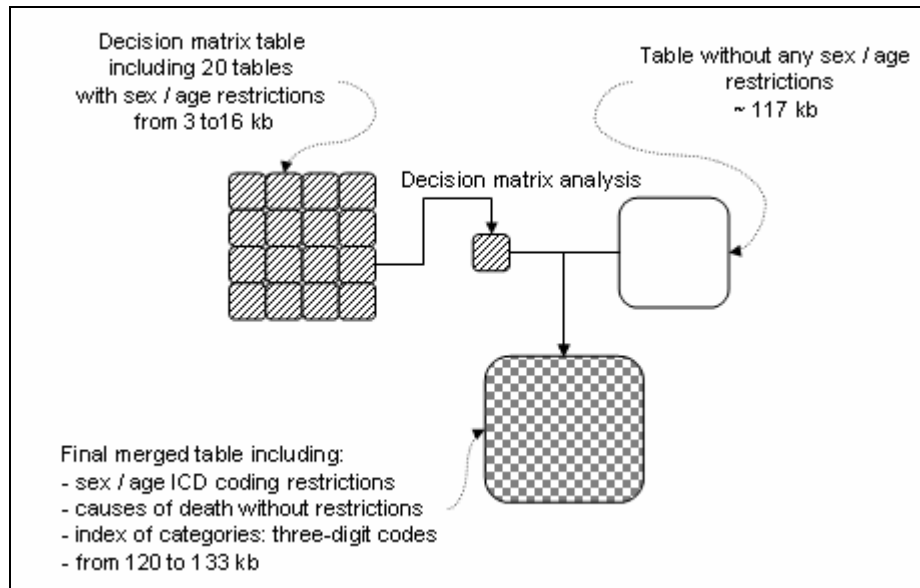
- 1 Errors with sex / age restrictions and underlying cause of death - Because of the features of the ICD analyzer, the accurate selection of a specific table for each combination of sex and age is possible. This controlled method avoids future problems with sex and age because it is based upon the ICD coding rules for underlying causes of death with sex / age restrictions.



- 2 Waste time on irrelevant ICDL information - Via the choice of smaller tables, the ICDL is greatly compacted. Only the relevant ICD information is loaded which permits time-saving for the search into the ICDL information.
- 3 Inappropriate memory allocation - If the entire ICDL were to be loaded, one would expend more computation time and use more memory compared to loading of only relevant parts of the ICDL. For example, the maximum size of the merged table using this feature would range from 120 kb to 133 kb. Note that 95% of this size is given by the table 22, the table without sex or age restrictions. Figure 4.1.3

Figure 4.1.3

Size of the parts of the ICDL



## 4.2 The semi-speller

The significance of this algorithm is that it utilizes the ICD analyzer to avoid spelling correction by using the IC. Indeed, the advantages of spelling correction are accompanied by a time-saving correcting the misspelled words.

Spelling correction is an important task in an integrated parsing environment. The spelling correction in an integrated parsing environment is not like the word processors that submit a few candidate words to the user for selection. Spelling-correction must provide an automatic correction mechanism and the response should be fast. The two basic steps in spelling correction are: (1) detection of the error, and (2) automatic correction.

In this analysis, the detection of spelling errors is limited to the IC. When the index of diseases analyzer cannot find a matching character in the IC for an input word, it is considered misspelled and immediately it is auto-completed or corrected.

#### 4.2.1 Semi-speller methods

Useful and realistic spelling correction cannot be done in isolation.

First, the erroneous word is compared with the words in the IC. Second, the words that match with a minimum number of corrections are considered as candidates for replacement. Third, ranking of the candidates is done by sorting the error distance (how many letters of the misspelled words have similarity order to arrive at the candidate word). Finally, the incorporated letter that created the misspelled word is eliminated so as to follow the structure of letters in the IC.

The following processes are involved: Figure 4.2.1

1. Choice of words from the IC: This process decides what words of the lexicon are chosen and in what order they are compared with the erroneous word. For the IC (2,073 codes for the Spanish version), a simple loop through all words from the lexicon suffices.
2. String comparison algorithm: This is the process that, given the erroneous word  $W1$  and the word from the IC  $W2$ , makes the minimum number of similar letters in  $W1$  to be equal to  $W2$ . This comparative string algorithm is the fundamental process that drives the Semi-speller tool. It is the most important part of the semi-spelling correction since the rest of the system reacts to the result of this process.
3. Ranking: As the words from the IC are compared with the misspelled word, the number of matched characters is saved. For each mismatched character, an error value is assigned. The matching words are ranked with respect to the error value.
4. Automatic correction: The previous steps select the possible candidates for correction. This level goes beyond of the simplest correction. When the index of diseases analyzer finds a mismatched character, it eliminates automatically the character and returns the character position to the last position.

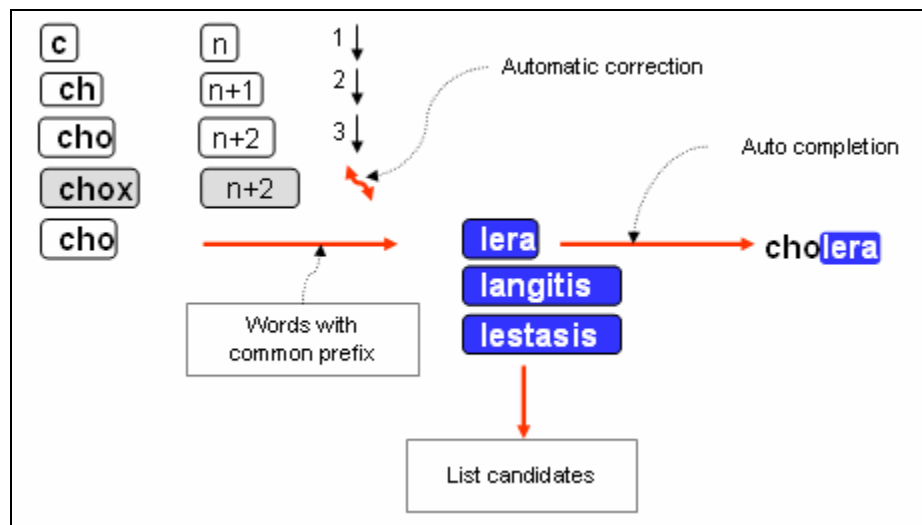
For example, for the word “*Cholera*”, which has seven letters, only requires that its first three letters be selected for ‘Cholera’ to be a possible candidate, but in the real world there are some mistyping errors. As demonstrated in the figure 4.2.1, the three first letters, ”*cho*”, are correctly typed, followed by a mistyping with the fourth letter “*x*”. In this case, the system automatically

corrects the word “*chox*” erasing the misspelled letter “*x*” and showing the list of candidates with the common prefix “*cho*”.

At the end, the system not only corrects the misspelled word and shows the list of possible candidates but it also ‘auto-completes’ the common prefix with the rest of letters of the best candidate.

Figure 4.2.1

Schema about the semi-speller correction method



In general, the first and the second step require further optimization when dealing with an IC with a large number of words. A review of the pertinent literature, confirms that the concept of edit distance has some similarities to the Semi-speller method. [Wu et al., 1992] The Semi-speller method extends the concept of edit distance by completing the whole word or phrase. Instead of counting any type of error correction as a one-error distance, the rest of the whole word is added for each candidate. This method permits one to avoid the whole process of having to complete the spelling of words that are long and often misspelled.

Generally, an input word that cannot be matched with any word from the IC could be in any of the following categories:

1. General spelling errors: the user does not know the correct spelling for the initial part of the word
2. Special characters: the user types numbers or special characters such as \*, \$, %, etc.

3. Space: the user types a space before the first character of the word, in which case it is eliminated automatically.

#### 4.2.2 Algorithm for comparing two words

The Semi-spelling correction is sequentially based on a character-by-character comparison of the erroneous word with a word from the IC. To reduce the time spent on comparing the erroneous word with irrelevant words from the ID, the number of mismatches is intentionally limited. This strategy follows the algorithm of Baeza-Yates. [Baeza-Yatez et al., 1992] Thus, this exemption permits one to save a lot of time comparing the misspelled word with the entire ID.

The Baeza-Yates' algorithm is based on finite automata theory, as the Knuth-Morris-Pratt [Knuttt et al., 1997] algorithm, and it also exploits the finiteness of the alphabet, as in the Boyer-Moore algorithm [Boyer et al., 1977]. In this algorithm, we assume that

$pat$  is a pattern of length  $m$ ,

$text$  is text of length  $n$ ,

$m$  is a vector of different states, where  $i$  tell us the state of the search between the positions  $1, \dots, i$  of the pattern and positions  $(j - i + 1), \dots, j$  of the text, where  $j$  is the current position in the text,  $s_i^j$  is the set of states (for  $1 \leq i \leq m$ ) after reading the  $j$ -th character of the text.

Assuming that  $b$  bits are necessary to represent each individual state  $s_i$ , the vector state can be represented by:

$$state_j = \sum_{i=0}^{m-1} s_{i+1}^j 2^{b-i}$$

For instance, for string matching we need only 1 bit (that  $b = 1$ ), where  $s_i$  is 0 if the last  $i$  characters have matched or 1 if not.

The same algorithm solves mismatches efficiently. For example, if  $k$  is the number of characters of the pattern to mismatch with corresponding text, the number of bits that are necessary to count  $k$  mismatches is

$$b = \lceil \log_2(k + 1) \rceil + 1$$

The aim here is to terminate comparisons of the misspelled word with the irrelevant words from the ID as soon as possible. For instance, comparison of “*Cholera*” and “*Shigellosis*” is terminated after mismatch of “*cho*” and “*shi*”, and the process concludes that “*cholera*” is not a good

candidate for “*shigellosis*”. As the two words are compared character by character, the error value is assigned. As soon as the error value reaches the disagreement limit the process is terminated.

With the implementation of this fast algorithm and the selection of a lightweight table, the comparison of two words is improved. However, the comparison algorithm should also permit detailed auto-correction steps to be applied to the misspelled word to transform it to the word that it expected. In fact, the comparison of two words is key to the development of the auto-completion function.

The following figure 4.2.2 is a pseudo-code implemented in LISP that works with a string and returns the longest prefix to words in the argument list. There is a generator that when given a string STR and a list LIST, finds the longest completion of STR that is in LIST or all possible completions in LIST, or both depending on result-type: ‘string = longest completion ‘list = all the completions.

Figure 4.2.2

Pseudo code of the auto completion function implemented in LISP

```

;;The function that lists the words with common prefix
(defun common-prefix (words)
  (cond ((null words) nil)
        ((null (cdr words)) (car words))
        (t (reduce #'common-prefix-2 words))))

;;The function that finds the longest common prefix to words
(defun common-prefix-2 (word1 word2)
  (let ((mismatch (mismatch word1 word2)))
    (if mismatch (subseq word1 0 mismatch)
                sequencel)))

;;The function that auto completes and returns the list of candidates
(defun autocomplete (str list &key
  (generator #'identity)
  (key #'identity)
  (result-type 'string)) (let ((result nil)
  (result-minimal nil)
  (completions nil)
  (length (length str)))
  (dolist (item (funcall generator list))
    (case result-type
      (list (nreverse completions))
      (string result)
      ((t (values result (nreverse completions))))))
    (let* ((item-string (funcall key item))
           (mismatch (mismatch str item-string))
           (when (or (not mismatch) (= mismatch length))

```

```

                (when (and (not result-minimal) (member result-type '(t
string)))
                (cond (mismatch (if result (setq result (common-prefix-2
result item-string))
                (setq result item-string))
                (when (string-equal result str) (setq result-minimal
t)))
                (t (setq result str) (setq result-minimal t))))
                (when (member result-type '(t list))
                (push item-string completions))))))

```

#### 4.2.3 Assignment of error values

A mismatch or disagreement method is used to terminate the comparison of an irrelevant word with the input word. If the method is too tolerant, a large number of irrelevant words that could affect the system efficiency would result. For this reason, in order to have a wide acceptance capacity, the Semi-speller system is essentially based on an analysis of the beginning of the words.

The basic fundamental idea is to have an adaptive method that can dynamically look only for better candidates. Such a method can be adaptive only if it dynamically lowers the tolerances for errors as better replacement candidates are found.

As the misspelled word and the word of the IC are compared character-by-character for each type of character error, an error value is assigned. The error value is used to assist in the selection of the best matching word between the two candidates. The value 1 (yes) is for the character case.

Also, in error assignment, the Semi-speller algorithm permits one to distinguish amongst acute vowels, spaces and special characters. This facility greatly assists in the recognition of early errors and word boundary problems. Character substitution in situations of keyboard effects (adjacent characters), as well as similar sounding characters, is also part of the algorithm for the assignment of errors.

The error categories include in the Semi-speller method include:

1. Character case – Most likely this is a keyboard effect or the user is trying to emphasize a point. In addition, grammatical rules like capitalization of a word in the initial part of a sentence are representative of this kind of error, if indeed it is considered an error at all.
2. Acute vowel – Most likely the user does not utilize the correct semantic rules for Spanish grammar.

3. Missing character(s): Competence errors - Most likely the user does not know the correct spelling. (This error does not include keyboard effect wherein the user pushes a key softly and the character does not get inserted).
4. Added character(s): Competence errors: Most likely the user does know the right spelling (other than possible keyboard effects when the user pushes a key by mistake, or the adjacent key to a character is also pushed, or a key is pushed for too long so that the character is repeated).
5. Character substitution(s): Most likely a keyboard effect, or competence error.
6. Reversed order: Most likely a keyboard effect.

All of these error considerations help in the recognition of errors that occur at the beginning of the words. In this sense, as I will not develop a complete speller, I did not consider the following error categories:

1. Missing character(s) at the end of words: There are four possibilities.
  - a. Abbreviation: The user has typed ‘spec’ for ‘specification’
  - b. Word boundary problem: An extra white space is added in a word, resulting in a split. For example, ‘tubercu losis’ is typed for: ‘tuberculosis’.
  - c. Silent character: If the missing character is a silent one in the word from the ID, the error value is reduced.
  - d. A simple case of missing characters.
2. Added character(s) at the end: Extra character(s) at the end can signal a word boundary problem. The user types ‘CholeraVibrio’ in place of “Cholera Vibrio”.
3. Similar sounding character substitution.

A summary of the above categories and the Semi-speller intervention is given in table 4.2.3

Table 4.2.3

General spelling errors categories by our support

<b>Error type</b>	<b>Erroneous word</b>	<b>Matched word</b>	<b>Support</b>
Character case	Lung	Lung	Yes
Acute vowel and Spanish grammar	Patogénesis	Patogenesis	Yes*
Reversed order	Lugn	lung	Yes
Missing character(s)	Chlera	cholera	Yes
Missing characters(s) at the end	Choler	cholera	No
Added character(s)	Chxolera	cholera	Yes
Added character(s) at the end	Cholerax	cholera	No
Character substitution(s)	Cholira	cholera	Yes

Similar sounding character substitution	<b>Kool</b>	cool	No
---	-------------	------	----

\* This feature works in accordance to Spanish grammar.

#### 4.2.4 Semi-speller optimization methods

As previously discussed, there are two prerequisites for spelling correction:

- A collection of words to compare the erroneous input with
- A function that compares the two words.

To get the best-matched word result from the spelling correction, a comparison of the erroneous input word with all words in the ICDL is desirable. However, since this process costs time, this method becomes impractical. Hence it is necessary to adopt a method that reduces the number of words from the lexicon for comparisons.

More often than not, the initial character of the misspelled words is not part of the error and the lexicon size does not grow. To exploit this characteristic, the search in the IC starts at the beginning of those words that have the same initial characters as the misspelled word. If acceptable candidate(s) within the first character are found, then one does not need to continue the search within other segments of the IC.

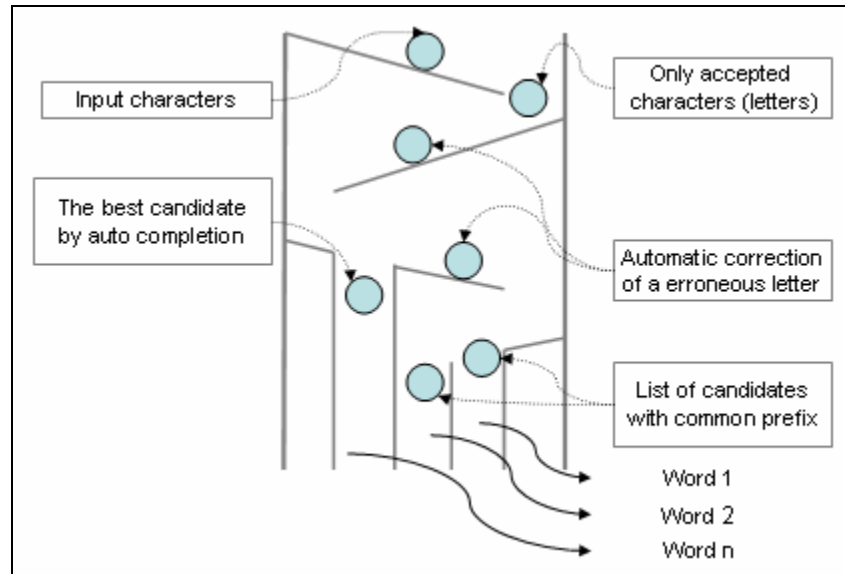
This method is extended to adjacent characters, beginning with the first character, in a loop form until an acceptable group of words is identified. If this method fails in the segment with the same initial character, the bad character that made an error is deleted in order to follow the search or follow the auto-completion. Of course, this method conserves the initial segment and continues the search character by character.

The Semi-speller presented here is not solely dependent on the first letter. It only starts the search from the section of the IC having the same initial character as the input word. This issue could be a big deficiency, but it can potentially reduce a large search to a small one, since in most cases, when a close candidate is found, the search is terminated and the ICD-10 has a limited IC. Figure 4.1.4

Figure 4.2.4

Representation of the semi-spelling correction





### 4.3 ICD analyzer

In this part, I will discuss how the ICD analyzer works with the sex and age inputs and how it deals with the cause of death.

The work of the ICD analyzer has focused on three areas: First, there has been an effort to understand the mortality coding workflow. Second, new methods to improve its coding speed, reduce its errors, and apply it to new contexts have been implemented. Third, new methods of general automated coding and clerically assisted computer coding have been investigated.

The ICD analyzer will be described in more detail below. A description of each algorithm under consideration will be included with the results of the analysis of that algorithm.

#### 4.3.1 Decision matrix analysis

As explained above in the part of structure and access of the ICDL, the ICD analyzer analyzes the table of decisions named DMT. This analysis, representing all the ways that the ICD analyzer can make a decision, is used to select a range of specific causes of death. In fact, the ICD analyzer makes decisions by comparing the inputs fields with pre-coded categories, ranges and numbers in the DMT. The input fields that supply the keys of sex and age are the parameters that determine what group of causes of death the user will use to search for the ICD information.

Given this methodology, the implementation of twenty discrete tables for sex and age inputs into the ICD analyzer minimizes the potential for basic errors in the assignment of codes. The

previously-discussed decision matrix analysis permits the system to increase its accuracy and performance because the coding workflow direction, used with other modern algorithms [Appel, 1991] is utilized.

#### 4.3.2 Cause of death phrase analysis

Using functions of the semi-speller the ICD analyzer performs a cause of death analysis. This analysis includes the selection of a category of disease from the index of diseases.

As previously discussed the cause of death is the best ambiguous input; in this sense, in the selection of disease categories the Semi-speller, crucial to handle lexical ambiguities, gives assistance for the ICD analyzer.

Also, as mentioned at the semi-spelling correction part, the spelling error detection is limited to the beginning of misspelled words. When the next input letter cannot be located in the sequence of the word, the system assumes a spelling error. The spelling correction process both finds those words in the ICDL that meet the minimum disagreement threshold with the misspelled word, and auto completes the rest of the word using the same prefix. The ICD analyzer might eliminate some of the candidates using the mortality coding rules but it is not necessary since the decision matrix analysis has already solved this problem.

Before the next process in which the ICD analyzer expands the cause of death selection, the ICD analyzer, using string tokenization and the Semi-speller functions, fetches the cause of death for a group of possible categories. For example, a user having in mind the cause of death “*Cholera by Vibrio cholerae*” should search for the ICD code using the first word “*Cholera*” which includes all the diseases involved with cholera. As one can see in the table 4.3.2 in which “cholera” includes three specific diseases.

Table 4.3.2

Relationship between the category of disease and the specific diseases

Category of diseases (ICD3)	Specific diseases – complete description (ICD4)
Cholera (A00)	Cholera caused by <i>Vibrio cholerae</i> 01, biotype cholerae (A00.0)
	Cholera caused by <i>Vibrio cholerae</i> 01, biotype El Tor (A00.1)
	Cholera no specified (A00.9)

#### 4.3.3 ICD coding

In the final analysis, the selected category is searched into the ICD4 containing the complete description in order to get the complete ICD information. In this final step the ICD analyzer only

reveals the related specific diseases of a selected category. For instance, if a user had selected the category “*Cholera*”, the system would have expanded this category showing the three specific diseases related to “*Cholera*”, as demonstrated in Table 4.3.2.

#### **4.4 IcodeX in action**

The IcodeX system was provisionally implemented in a simple graphical user interface (GUI) because this coding tool will be a plug-in such as an archive DLL or Active X, instead of a complete software program. In the appendix B, there are some screenshots about the IcodeX system in action using a provisional GUI.

In fact, IcodeX will be a plug-in and IcodeX currently has the status of prototype. In the future IcodeX will be used by developers to develop integrated and complete software programs which need to deal with mortality coding in speaking Spanish countries. With the status of prototype, IcodeX works very well; however, it needs more evaluation. The next is the examination of IcodeX.

The prior detailed description of the system components is incomplete without an analysis and examination of the interaction amongst the system components. In completing this analysis, a few situations of the process are presented to demonstrate the operation of the system.

##### **4.4.1 Cooperative processes in the cause of death coding**

The interleaved component processes:

1. Decision matrix analysis
2. Semi-spelling correction
3. Cause of death analysis
4. Expanded coding

work together to find the ‘best’ matching code from the ICDL, given an entry of an ambiguous cause of death.

The sequential steps for mortality coding and the choice of the best candidates are organized and shown as follows:

- 1** The ICD analyzer seeks errors in the inputs of age and sex
- 2** The ICD analyzer utilizes the DMT and these inputs to select a specific table of the ICDL
- 3** The ICD analyzer seeks to identify basic errors in the first letters of the word entry for the cause of death (prefix)

- 4 The ICD analyzer passes the prefix to the Semi-speller
- 5 The Semi-speller returns a list of one or more possible categories for the prefix. The list is sorted in ascending order.
- 6 The Semi-speller auto-completes the prefix of the best candidate. If the phrase can be auto-completed, this category is considered the only candidate for the prefix and is passed to the ICD analyzer. Otherwise,
- 7 The ICD analyzer waits the selection of other candidates of the category list
- 8 The ICD analyzer examines the selected category and expands this category into its sub-list
- 9 The ICD codes the selected candidate from the sub-list

#### 4.4.2 Basic and complex coding – Two main contexts

Health coders sometimes have difficulties coding health data. Amongst these coding problems there exist basic and complex coding contexts.

As demonstrated in the figure 4.4.2, there are seven common situations in which IcodeX solves each one of these problems. These coding situations have been segregated into two parts: ‘Simple’ and ‘Complex’.

- Simple coding: This group includes the situations “a”, “b” and “c”.

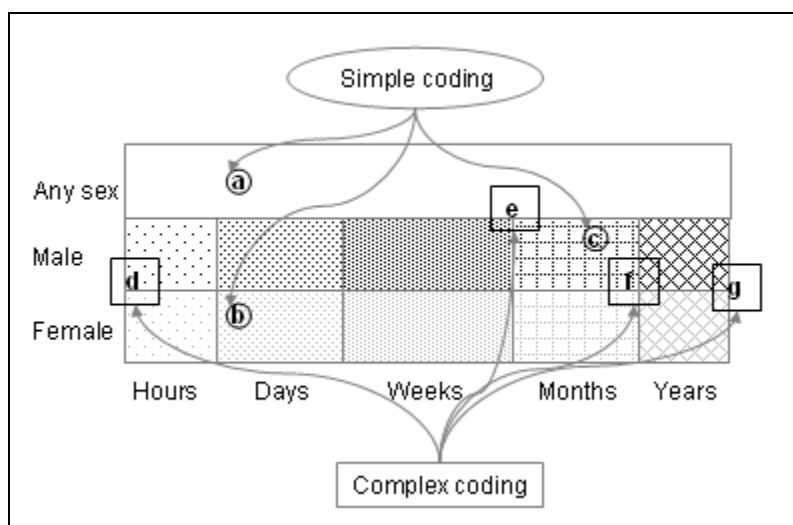
- **Situation a:** the health coders code mortality data without age and sex restrictions as death by trauma.
- **Situations b, c:** The health coders code mortality data with understandable sex restrictions as death by cancer of prostate or death by breast cancer.

- Complex coding: This group includes the situations “d”, “e”, “f” and “g” which are not easily understandable the sex or age restriction.

- **Situation d:** The health coders code mortality data with minimum range of age restrictions such as death by neonatal cancers.
- **Situation e:** The health coders code mortality data with no understandable sex restrictions mixed with some sort of age restriction between weeks and months.
- **Situation f:** The health coders code mortality data with no understandable sex restrictions and age restrictions, the most complex of the examples.
- **Situation g:** The health coders code mortality data with no understandable maximum range of age restrictions.

Figure 4.4.2

Simple and complex coding solved by the coding tool



The seven coding contexts were tested running IcodeX on a Pentium III, 1000 MHZ. Although some causes of death need more letters to catch the ICD-10 code than other causes of death, the memory usage and CPU time uniformly were less than 1 second and were similarly fast in the IcodeX analysis.

#### 4.4.3 Coding rules of frequency

Since I am applying the ICD coding rules, IcodeX will not produce any coding errors. But, a novice user can find suspicious or incoherent codes because I am not integrating the frequency coding rules. As a consequence, a user should code abort as cause of death in a child since the ICD-10 rules permit us to use this code. This part will be solved with the next part of the project together with the evaluation and the integration of the tool.

Given that this analysis is based upon the mortality data from Peru and that this methodology is based upon an analysis of the most frequent causes of death in Peru, it is apparent that the methodology is optimized for the Peruvian population. Given that other countries likely have specific mortality characteristics, it is possible that this methodology is not optimized for use in other countries. I did not integrate the referred list of probable causes of death because it follows the same situation of the frequency coding rules. In this sense, I am only integrating the ICD coding rules with accuracy.

In summary, this chapter describes the design and development of IcodeX for coding basic cause of death. IcodeX is a mixture of parsing and semi spelling techniques; as a result, it is a fast coder that handles ambiguous causes of death.

An fast coder requires that errors be detected without user intervention. In this sense, the semi spelling correction and the ICD analyzer work hand and hand: the ICD analyzer gets the input and passes it to the semi speller and all of these coding actions are quickly performed because I organized the Lexicon in order to get accuracy and speed coding a cause of death.

IcodeX is organized in three modules (ICDL, ICD analyzer, semi speller). The first module, the ICDDL was set up to facilitate the functions of the semi speller and the ICD analyzer, the third module, guarantees accurate and fast access to the ICD-10 codes.

## CHAPTER 5

### 5. Summary of contributions and future research directions

Chapter 5 summarizes the contributions of this thesis project to the public health informatics community especially for Peruvian policymakers. It also provides a few suggestions for future research directions.

#### 5.1 Summary

There is no doubt that the quality of making health policies has a strong relationship with the quality of health data and vital statistics. A clear benefit of the IcodeX solution is that it will work to improve the quality of vital statistics in Peru with a direct consequence that it indirectly would help Peruvian policymakers and governmental health agencies.

Critics are bound to oppose this solution, claiming that it is either too impractical or too difficult to implement. However, one only needs to look to the American National Center for Health Statistics (NCHS) to realize how the implementation of successful NCHS coding solutions have served to avoid coding errors in the USA and the rest of the world. Considering how the health statistics agencies have been aware of the need to solve the medical standardization problem, this solution certainly appears worthy of a trial. When such programs were suggested many years ago, health coders and public health people scoffed at these initiatives. However, today there are important standardized medical vocabularies and coding rules and automated coding programs that have dramatically improved the quality of vital statistics in some developed countries. A tool such as the IcodeX that requires minimal training would appear to have great potential.

Several solutions have been suggested for solving errors with mortality coding. However, those solutions are often only self-instruction systems or ICD-10 training materials. In fact, PAHO has encouraged the use of INTERCOD for Spanish-speaking codes; WHO-Europe has promoted RUTENDON for Russian health coders, and WHO-India has encouraged the use of ACBA for Indian health coders. The use of these materials is a very problematic because these approaches do not address health coding problems such as medical vocabulary and human errors. The other problem with these 'solutions' is that they simply will not provide the profound changes that are necessary on the part of the health coders when errors occur.

IcodeX is not a tool for simply searching codes. IcodeX is a tool that incorporates complex rules, and in fact, the design of the IcodeX tool deliberately incorporated the complex ability to handle mortality-coding requirements, as specified above. In fact, IcodeX meets the challenges facing coding errors via its combination of fast spelling correction and searching methods within a segmented lexicon.

The IcodeX solution will put an end to one of the most significant coding problems faced by PAHO members with respect to mortality data, i.e., the challenge of maintaining good vital statistics using mortality data. In point of fact, PAHO has currently shown its interest in mortality data, and it needs some basic health informatics support. Therefore, IcodeX represents an excellent opportunity to begin a sustainable information system support.

The ministry of health in Peru will be the best place for testing IcodeX because it is the main caregiver and the principal health policymaker in Peru. There is a lot of work that will be required to apply and successfully implement a trial for the integration of IcodeX into the governmental framework. This future work will provide experience with the integration of IcodeX into other health systems of other PAHO members. In fact, the next step and the primary focus of IcodeX is a rigorous evaluation in the real world to assess its accuracy and performance.

The future integration of IcodeX into the coding health workflow of PAHO members has similar problems and limitations. Hence, the problem with mortality data and vital statistics can be generalized to all the PAHO members, and IcodeX probably will deal with comparable problems in the subsequent phase of implementation.

## **5.2 Future research directions**

Although the emphasis of this analysis has been the mortality coding errors in Spanish for an implementation in Peru, there are lots of unknown or undefined problems that very likely would be managed with the IcodeX system. At the current time no 'real-world' implementation of IcodeX exists. The first step in the future research directions certainly should include an intense evaluation of IcodeX in the real world. As part of this evaluation, it is anticipated that IcodeX will be tested in Peru at some time in the near future so that IcodeX might ultimately be implemented in the health system.



In the end, the coding algorithm should be improved and debugged using the information resulting from these first evaluations.

IcodeX currently is a prototype. Since IcodeX will be a plug-in, the research study of IcodeX does not care with usability issues. Consequently, the focus of this research has been to develop a robust plug-in that might be integrated into any type of software that requires coding mortality data using the ICD-10.

The IcodeX solution was designed for ICD-10, but the principles on which it was designed might apply similarly to other controlled vocabularies. Many medical rule-based systems potentially have similar constructs and problems to ICD-10. In this sense, a likely first extension of IcodeX would be to expand the ICD-10 implementation beyond 'mortality', possibly incorporating health-business and health-statistics applications that might require ICD-10 interfaces. There are undoubtedly other primarily non-healthcare implementations that might benefit from integration with ICD-10.

A successful implementation of IcodeX in the ICD-10 world would certainly merit the evaluation of some of these other systems to assess whether reasonable simple modification of existing IcodeX features might permit implementations in other contexts. The Semi-speller feature of IcodeX requires no significant changes at all since it is designed to deal with any set of word entries from any given lexicon, for instance. If a standard Latin-character controlled vocabulary were available in an electronic database format, it is evident that the 'Semi-speller' could be successfully implemented. It is even possible that the Semi-speller option could be a plug-in to any software system that has an accessible dictionary and a word editor.

Finally, I am very thankful to have had this opportunity to develop and improve the prototype of IcodeX, and I am hopeful that I can find opportunities to implement it so that it might demonstrate great significance and impact upon health policies in WHO members.

## BIBLIOGRAPHY

1. Anderson RN, Miniño AM, Hoyert DL, and Rosenberg HM. Comparability of cause of death between ICD-9 and ICD-10: Preliminary estimates. *National vital statistics reports*; vol. 49 no. 2. NCHS. 2001.
2. Appel, M. V. and Hellerman, E. (1983). "Census Bureau Experiments with Automated Industry and Occupation Coding," *Proceedings of the American Statistical Association*, 32-40.
3. Appel, M. V. (1991). "Field Weights for the Automated Coder", Memorandum to John Priebe, Census Bureau, Washington, DC, July 25.
4. Arialdi M, Minimo Harry M. Rosenberg, Eds. *Proceedings of the international collaborative effort on automating mortality statistics*, Vol II. 2001.
5. Boyer R, Moore S. A fast string searching algorithm. *CACM*, 20:762-772, 1997.
6. Brown P, O'Neil M, Price C. Semantic representation of disorders in Version 3 of the Read Codes. *National Health Service Centre for Coding and Classification*, Loughborough, United Kingdom. 1999.
7. Bechhofer S. GRAIL frequently asked questions, *GALEN Documentation C1* (University of Manchester, Manchester, U.K., October, 1994)
8. Bernauer J, Subsumption principles underlying medical concept systems and their formal construction, in: J.G. Ozbolt, ed., *Proceedings of the Eighteenth Annual Symposium on Computer Applications in Medical Care*. 1999.
9. Brachman R.J, Fikes R E and Levesque H J. KRYPTON: A functional approach to knowledge representation. *IEEE Computer* (October, 1983) 67–73.
10. Brachman RJ and Schmolze J G. An overview of the KL-ONE knowledge representation system, *Cognitive Science* 9 (1985) 532–539.
11. Brachman R J, McGuinness D L, Patel-Schneider P F, Resnick L A and Borgida A. Living with CLASSIC: when and how to use a KL-ONE-like language, in: J. Sowa, ed., *Principles of Semantic Networks: Explorations in the Representation of Knowledge* (Morgan Kaufmann, San Mateo, CA, 1991) 401–456.
12. Brett A. New guidelines for coding physicians' services – A step back. *N Engl J Med* 1998; 339:1705-8.
13. Blanquet A, Zweigenbaum P. A lexical method for assisted extraction and coding of ICD-10 diagnoses from free text patient discharge summaries. *Service d'Informatique Medicale/DSI/AP-HP & Biomathematiques U. Paris* 6. 1990.
14. Bureau of the Census. USA 1990.
15. Baeza-Yatez R, Gonnet G. A new approach to text searching. 1992, *Commun. ACM*, Vol.35, No 10 (Oct), 74-82.
16. Chirinos J, Soldevilla L, Alcantara E. Morbidity and maternal mortality from Peru. *Peruvian Journal of Epidemiology*, 1994 – Vol. 7 No 1 July.
17. Cole R. A medical text classification agent using SNOMED and formal concept analysis. *Department of Computer Science. Universidad of Adelaide*. Nov 1995.
18. Chin L, Krall M, Lester S. ICD9 and SNOMED: Adapting Clinical Coding Systems for use in the *Computer-Based Patient Record*. 2001.
19. CDC, NCHS. *Proceedings of the international collaborative effort on automating mortality statistics*, Vol I. 1999.
20. CDC-NCHS. International Collaborative effort on automating Mortality statistics, 1st Wa D.C. *Proceedings of the International Collaborative effort on automating Mortality statistics*, 1996.
21. Chamblee, R.F. and M.C. Evans, Transax: The NCHS Multiple Cause of Death Axis Translation System for 1968-1978 *Mortality Data, Vital and Health Statistics*, Series I. to be published, DHHS.

22. Cleone R. Automated coding in England. *Office for National Statistics*. 2001.
23. Cochrane Collaboration on Effective Professional Practice. The Cochrane Collaboration on Effective Professional Practice data collection checklist. York: *Department of Health Sciences and Clinical Evaluation*, University of York, 1996.
24. Dai Y, Lohn T. A new statistical formula for Chinese text segmentation incorporating contextual information. *SIGIR '99 Berkley, CA. ACM 1-58113-096-1/99/007*.
25. Dianne Miller. Automated Coding At Statistics Canada, *General Systems, Informatics Branch, Statistics Canada*, November 1991.
26. Frances A, Pincus HA, First MD (Ed). *Diagnostic and statistical manual*, Fourth Edition. American Psychiatric Association, Washington, D.C. 1994.
27. Franz P, Zaiss A. Schultz S. Automated coding of diagnoses three methods compared. *Department of Medical Informatics*. Freidburg University Germany. 1999.
28. Guzman A. Improving the ENAHO IV questionnaire of Health. 1996, improving the surveys about human condition in Latin American and the Caribe. *INEI – MECOVI. Technical document*. BID, BM and CEPAL.
29. Gruber TR. A translation approach to portable ontologies, *Knowledge Acquisition*, 5(2) (1993) 199–220.
30. Gibbons PS. Terminology II: Establishing the Consensus. Nat Conf on Terminology for Clinical Patient *Description*. VA 1999, Apr.27-29.
31. Goodaire, Edgar G., Paramenter, Michael M. *Discrete Mathematics with Graph Theory*. New Jersey: Prentice Hall, 1998.
32. Goldacre M, Roberts S and Griffith M. Multiple-cause coding of death from myocardial infarction: population-based study of trend in death certificates data. *Journal of Public Health Medicine*. 2003 vol.25, No 1, pp 69-71.
33. Gillman D, Appel M, Automated coding research at the census bureau. *Statistical Research Division*
34. Hsia D. Krushat W, Fagan A, Tebbutt J, Kusserow R. Accuracy of diagnostic coding for Medicare patients under the prospective-payment system. *N Engl J Med* 1988;318:352-5.
35. Hamming R. *Coding and Information Theory*, 2e, Prentice-Hall, 1986, p. 27
36. INEI. Dirección Técnica de Demografía y Estudios Sociales. Perú. *Proyecciones departamentales de población*.1995-2015. Lima, 1995
37. Jamouille M. ICPC use in the European community 16<sup>th</sup> WONCA *World Congress of family doctors* Durban, South Africa. 2001-2002.
38. Jablin, C. (1992). "Evaluation of Automated I&O Coding for Current Surveys", *Memorandum to Sherry L. Courtland*, Census Bureau, Washington, DC, October 15.
39. Johansson L. Swedish MIKADO coding system. *Statistics Sweden*. 2001.
40. Krall MA, Chin HL, Dworkin L, Gabriel K, Hayami D, Towery B, Wong R. Improving the Acceptance and Performance of Clinicians in the Diagnostic Coding Task Required by an Outpatient Computer Based Medical Record. *Am J Managed Care*. 1997: 3(4):597-601.
41. Kornai A, Stone L. Automatic translation to controlled medical vocabularies. *PPD informatics- research*. Cambridge – NCI-NIH. 2001.
42. Knuth DE, Morris J, Pratt V. Fast pattern matching in strings. *SIAM J on Computing*, 6:323-350, 1997.
43. Lussier Y, Shagina L, Friedman C. Automating ICD-CM encoding using medical language processing. A feasibility study. *Dep of Medical Informatics*, Columbia University, NY. Postdoctoral research. 2001.
44. Letrillart L, Viboud C, Boelle P. Automatic coding of reasons for hospital referral from general medicine *Free-text reports*. INSERM unit 444 – WHO collaborative center for electronics diseases surveillance, Paris, France. 2001.

45. Lussier Y, Shagina L, Friedman C. Automating ICD-CM encoding using medical language processing. A feasibility study. *Dep of Medical Informatics, Columbia University, NY. Postdoctoral research.* 1999.
46. Lovis C, Baud R, Rassinoux AM, Michel PA, Cherrer JR. Medical dictionaries for patient encoding systems a methodology. *Artif Intell Med* 14:201-214, 1998.
47. Mario Nascimento, Adriano da Cumba. An experiment Stemming Non-Traditional text [www.dcc.unicamp.br/~mario](http://www.dcc.unicamp.br/~mario). 1999.
48. MedDRA. *Introductory Guide*, version 4.1, MSSO-DI6003-4.1.0 2001
49. MeSH: Annotated Alphabetic List. National Library of Medicine, Bethesda, MD, 1997.
50. McCray AT and Nelson S J. The representation of meaning in the UMLS, *Methods of Information in Medicine*, 34 (1995) 193–201.
51. Mohammad Ali Elmi., (1992). A Natural Language Parser with Interleaved Spelling Correction Supporting Lexical Functional Grammar and Ill-Formed Input. *Ph.D. Dissertation, Computer Science Department, Illinois Institute of Technology, Chicago, Illinois.*
52. Macchia S, De Angelis R. Applying automated coding to the pilot survey of next population census: a challenge. *National Institute of Statistics, ISTAT. Methodological studies department.* Italy. 1999.
53. McCall, J., P. Richards, and G. Walters. 1977. Factors in Software Quality, NTIS AD-A049-014, 015, 055. November
54. NCHS: ICD-9 ACME Decision Tables for Classifying Underlying Causes of Death, 1984, NCHS *Instruction Manual, Part 2c.* Public Health Service, Hyattsville, Md., Aug.1983. National Center of Health Statistics. 1997.
55. NCHS – CDC. U.S. Vital Statistics System. Major Activities and developments 1950 – 1995.
56. Oliver D, Shashar Y, Shortlife E, Musen M. Representation of Change in Controlled Medical Terminologies Stanford Medical Informatics MSOB X-215 Stanford University School of Medicine Stanford, CA 94305-5479. 2000.
57. Pan American Health Organization. Health sector reform: *Proceedings of a special meeting.* ECLAC/IBRD/IDB/OAS/PAHO/WHO/UNFPA/UNICEF/USAID. Washington, DC, 1996 Sep 29-30.
58. PAHO. Second meeting to establish a surveillance network for Emerging Infectious Diseases (EID) in the Amazon Region. *PAHO/HCP/HCT/143*, 1999.
59. Pacak MG, Norton LM, and Dunham GS. Morphosemantic analysis of –IT IS forms in medical language. *Methods Inf Med* 1980; 19:99-105
60. Pacak MG, Norton LM, and Dunham GS. Morphosemantic analysis of –IT IS forms in medical language. *Methods Inf Med* 1980; 19:99-105.
61. Roger A, Cote, 1978. The SNOP-SNOMED Concept; Evolution towards a Common Medical Nomenclature and Classification. *Proceedings of the Seventh International Congress on Medical Records.*
62. Rowe E, Wong C. An introduction to the automated coding system. *Automated Coding Staff Statistical Research Division Bureau of the Census Washington, DC 20233.* 1990.
63. Rosen, Kenneth H. *Elementary Number Theory and Its Applications.* Ontario: Addison-Wesley. 1990.
64. Rada R, Computerized coding of the medical problem statement. *Baylor College of Medicine. Report – Grant NIH.* Houston Texas, USA. 1990.
65. Rowe, E., Wong, C. (1994). "An Introduction to the ACTR Coding System", *Bureau of the Census Statistical Research Report Series* No. RR94/02.
66. Slee V, Slee D, Schmidt H. The Endangered Medical Record. Ensuring its integrity in the age of informatics. Tringa Press Saint Paul, Minnesota. 2000.

67. Straub, H. Four Models of Concept Architectures. In: Grütter, R. (ed.): *Knowledge Media in Healthcare: Opportunities and Challenges*. Idea Group Publishing, Hershey/London (2002).
68. Saltzman A. Adverse reaction terminology standardization: A report on Shering-Plouh's use of the WHO dictionary and the formation of the WHO adverse reaction terminology users group. *Drug Information Journal* 1985;19:35-41.
69. SNOMED CT – first release. *Technical reference manual*. January 2002, College of American Pathologists. SNOMED – IL.
70. Statistical commission and Economic commission for Europe. Conference of European Statisticians. WHO regional office for Europe. *ECE-WHO meeting on Health statistics*. Rome, Italy, Oct 1998.
71. Slee. *ICD-10 Procedure Coding System*. Introduction, training manual and tabular list, HCFA, Internet, Summer 1998.
72. Thorsten K, Stoffel K. Going beyond stemming: creating concepts signatures of complex medical terms. *Knowledge-Based Systems* 15(2002) 309-313.
73. *UMLS Knowledge Sources*, Eighth Edition Documentation, National Library of Medicine, Bethesda, MD, 1997.
74. Verhoeff J. "Error Detecting Decimal Codes", *Mathematical Centre Tract 29*, The Mathematical Centre, Amsterdam, 1969.
75. Wolff S. Automating coding of medical vocabulary. In: Sager N, Friedman C, and Lyman MS, eds, *Medical Information Processing – Computer management of narrative Data*. Addison Wesley, Reading Mass, 1987:145-62.
76. World Health Organization 1992. *Manual for international classification of diseases and health related problems, 10<sup>th</sup> revision*. Geneva, Switzerland.
77. World Health Organization 1969. *Manual for international Classification of diseases, 8<sup>th</sup> revision*. Geneva Switzerland.
78. WHO. *ICD-10. International Statistical Classification of Diseases and Related Health Problems*. Geneva (Switzerland); 1992.
79. WHO. Telematics Policy in support of WHO's Health-for-All Strategy for Global Health Development. *Report of the WHO Group Consultation on Health Telematics*. 11-16 December 1997.
80. Weber, R. P. *Basic Content Analysis*, 2d ed. Newbury Park, CA: *Sage Publications*, 1990.
81. Wagner and Putter, "Error Detecting Decimal Digits", *CACM*, Vol 32, No 1-january 1989, pp 106-110.
82. Wagner and Putter, "Error Detecting Decimal Digits", *CACM*, Vol 32, No. 1 (January 1989), pp. 106-110
83. Wagner N, Putter P. Error detecting decimal errors. *Communications of the ACM*. January 1989(32) 1.
84. World Health Organization 1992. *Manual for international Classification of diseases and health related Problems*, 10th revision. Geneva, Switzerland.
85. Wu S, Manber U. Fast text searching allowing errors. 1992, *Commun ACM*, Vol. 35, No.10 (October), 83-91.
86. Yasnoff W, Overhage M, Humphreys B, et al. A national agenda for public health informatics. *J Public Health Management Practice*, 2001, 7(6), 1-21.
87. Zweigenbaum P, Grabar N. A contribution of medical terminology to medical language processing resource: Experiments in morphologic knowledge acquisition from thesauri. *Proc Conference on Natural Language processing and Medical concept representation*, Phoenix, Az, 16-19 Dec 1999.

## APPENDIX A

### ICODEX user manual

#### A.1 System requirements

IcodeX works with these operative systems: Windows 95, 98, NT, 2000 and XP.

#### A.2 Installation of the prototype

To install IcodeX, you must follow these steps

1. Click on the Setup of IcodeX
2. Select the directory to IcodeX files

#### A.3 Using IcodeX

To use IcodeX you must follow these steps.

1. After the installation, click on the shortcut of IcodeX.exe in the list of programs
2. Select the database CIE10.mdb. If you forget this step, IcodeX will do for you
3. Type the unity of age and the age in numbers on the text box for age. If you forget this step, IcodeX stop the coding process
4. Select the sex in the combo box, 0 for female and 1 for male. If you forget this step, IcodeX stop the coding process
5. Type the cause of death. You must type the characters at the beginning of the word in correct manner in order to get the ICD-10 code. If you mistype the rest of the words, IcodeX will spell the word and will get the best possible candidates
6. Select a category of cause of death. Remember that this is the category which contains the ICD-10 code that you are looking for. You can select the candidate using the down and up arrows or the mouse. To expand the category - double click or enter
7. Select the candidates of the expanded category. These candidates have the ICD-10 code that you are looking for. You can select the ICD-10 code using the down and up arrows or the mouse. To record the category - double click or enter
8. Record the inputs (Sex, age and ICD-10). Remember that IcodeX automatically restarts the system when you record a death certificate.
9. Close the IcodeX

#### A.4 Known bugs

IcodeX doesn't work with two possible reasons.

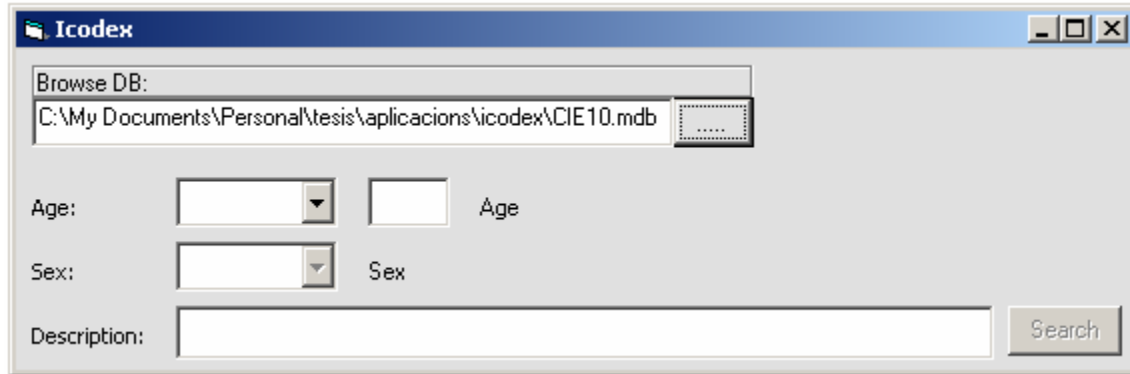
1. There is an error at the beginning of the characters of the cause of death
2. You are looking for a cause of death with the incorrect category

## APPENDIX B

### The ICODEX interface

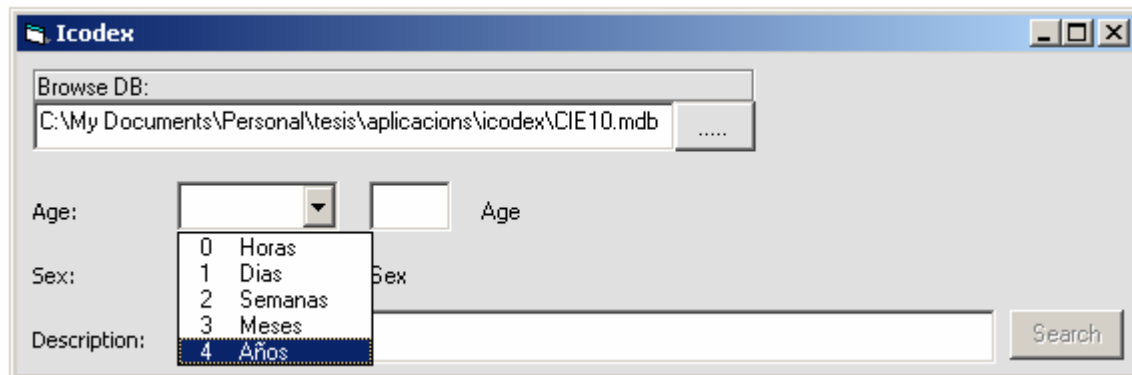
Step 1 – Starting IcodeX

Figure B.1 – Starting IcodeX



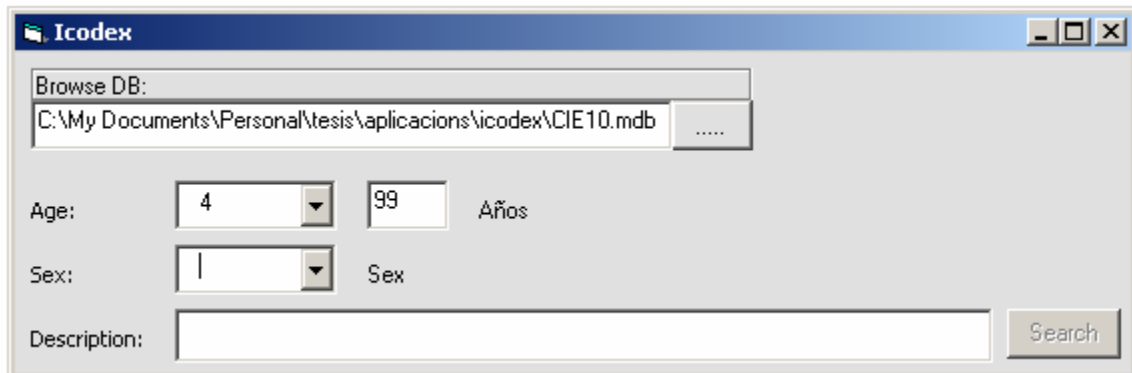
Step 2 – Coding the unity of the age.

Figure B.2



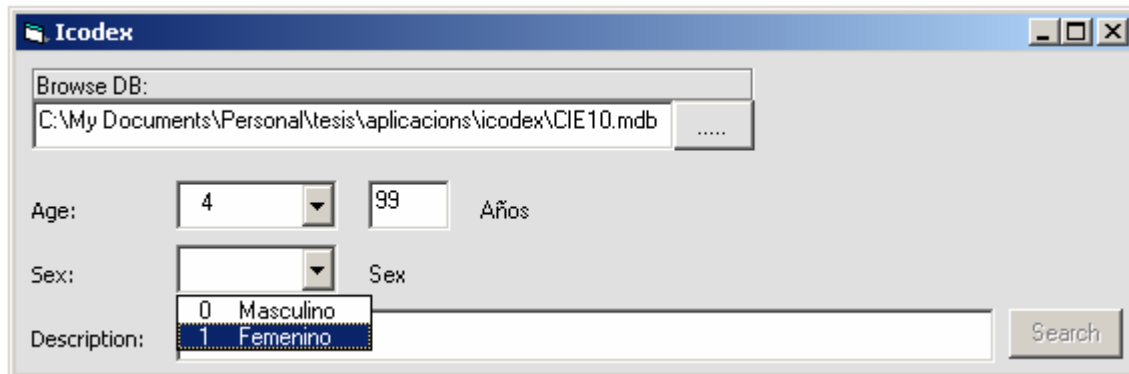
Step 3 – Coding the age (numbers).

Figure B.3



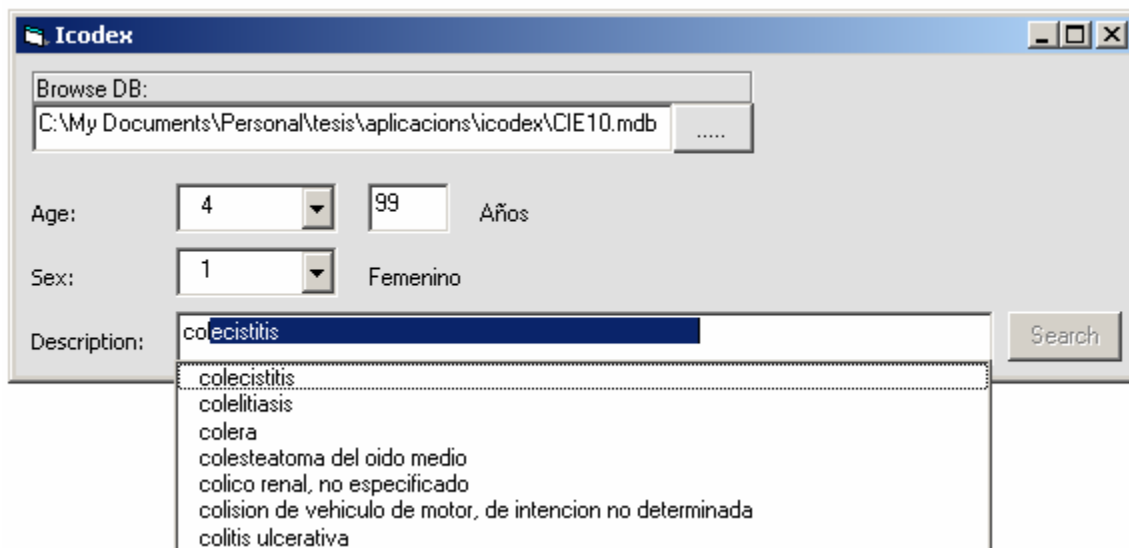


Step 4 – Coding the sex.  
Figure B.4



The screenshot shows the Icodex application window. The 'Browse DB:' field contains the path 'C:\My Documents\Personal\tesis\aplicacions\icodex\CIE10.mdb'. The 'Age:' field has a dropdown menu set to '4' and a text box containing '99' with the label 'Años'. The 'Sex:' field has a dropdown menu set to '1' with the label 'Sex'. The 'Description:' field has a dropdown menu with '0 Masculino' and '1 Femenino' options, where '1 Femenino' is selected. A 'Search' button is located to the right of the description field.

Step 5 – Coding the cause of death – Searching for the candidates and auto completing the best candidate.  
Figure B.5



The screenshot shows the Icodex application window. The 'Browse DB:' field contains the path 'C:\My Documents\Personal\tesis\aplicacions\icodex\CIE10.mdb'. The 'Age:' field has a dropdown menu set to '4' and a text box containing '99' with the label 'Años'. The 'Sex:' field has a dropdown menu set to '1' with the label 'Femenino'. The 'Description:' field contains the text 'colecistitis' and has a dropdown menu open showing a list of suggestions: 'colecistitis', 'colecistiasis', 'colera', 'colesteatoma del oido medio', 'colico renal, no especificado', 'colision de vehiculo de motor, de intencion no determinada', and 'colitis ulcerativa'. A 'Search' button is located to the right of the description field.

Step 6 – Coding the cause of death – Selecting the category of death.  
Figure B.6

The screenshot shows the Icodex software window. At the top, there is a 'Browse DB:' field containing the path 'C:\My Documents\Personal\tesis\aplicacions\icodex\CIE10.mdb'. Below this, the 'Age:' field is set to '4' and '99' with the label 'Años'. The 'Sex:' field is set to '1' with the label 'Femenino'. The 'CIE3:' field is set to 'A00'. The 'Description:' field contains the text 'colera'. A dropdown menu is open below the description field, showing a list of medical conditions: 'colecistitis', 'colelitiasis', 'colera' (highlighted in blue), 'colesteatoma del oido medio', 'colico renal, no especificado', 'colision de vehiculo de motor, de intencion no determinada', and 'colitis ulcerativa'. A 'Search' button is located to the right of the description field.

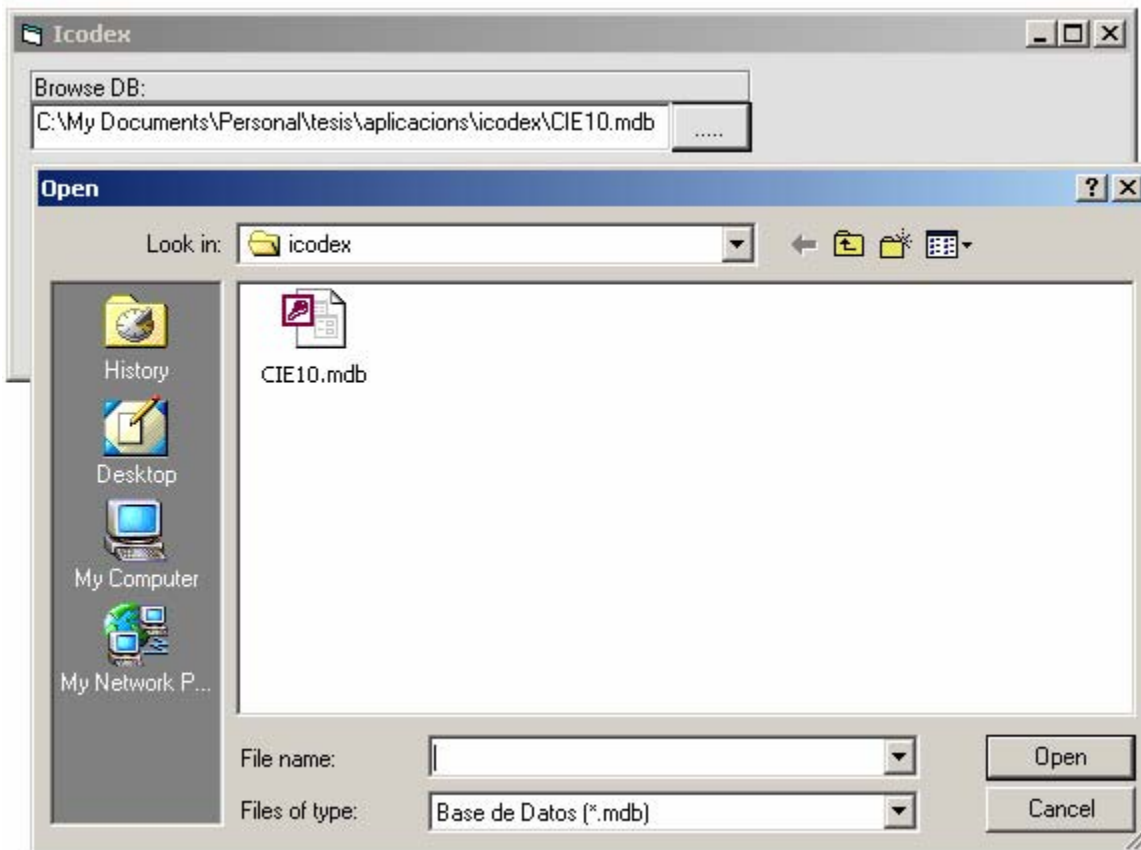
Step 7 – Expanding the cause of death selection.  
Figure B.7

The screenshot shows the Icodex software window with the same settings as Figure B.6. The 'Description:' field now contains 'colera' and a dropdown menu is open below it, showing a list of expanded search results: 'colera debido a vibrio cholerae o1, biotipo el tor', 'colera', and 'sospechoso de colera'. A 'Save' button is located to the right of the description field.

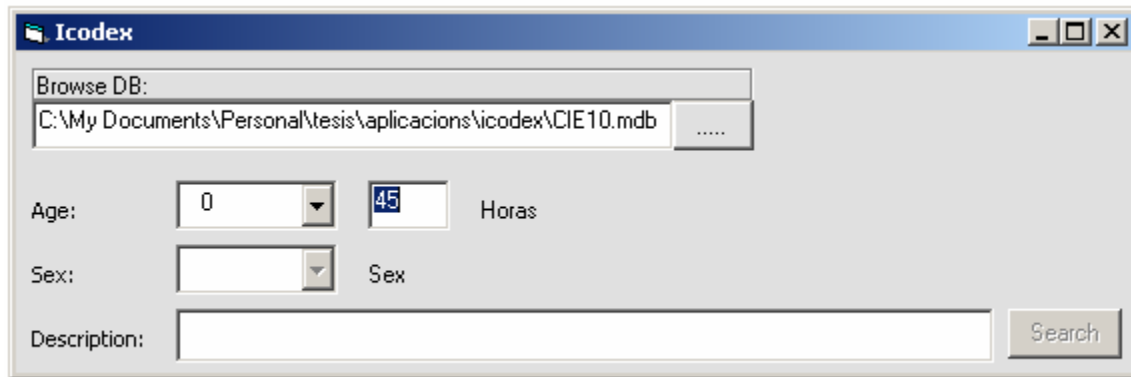
Step 8 – Coding the specific cause of death.  
Figure B.8

The screenshot shows the Icodex application window. At the top, the title bar reads "Icodex". Below it, there is a "Browse DB:" field containing the path "C:\My Documents\Personal\tesis\aplicaciones\icodex\CIE10.mdb" and a browse button. The "Age:" field has a dropdown set to "4" and a text box containing "99" with the label "Años". The "Sex:" field has a dropdown set to "1" and the label "Femenino". To the right, "CIE 4: A00.1" is displayed. The "Description:" field contains the text "colera debido a vibrio cholerae o1, biotipo el tor". A dropdown menu is open below this field, showing three options: "colera debido a vibrio cholerae o1, biotipo el tor" (highlighted), "colera", and "sospechoso de colera". A "Save" button is located to the right of the description field.

Optional steps:  
1 - Selecting another database.  
Figure B.9



2 - Sequence control and range of age  
Figure B.10



The screenshot shows a window titled "Icodex" with a search interface. At the top, there is a "Browse DB:" label and a text box containing the path "C:\My Documents\Personal\tesis\aplicacions\icodex\CIE10.mdb" with a browse button ".....". Below this, there are three search criteria:

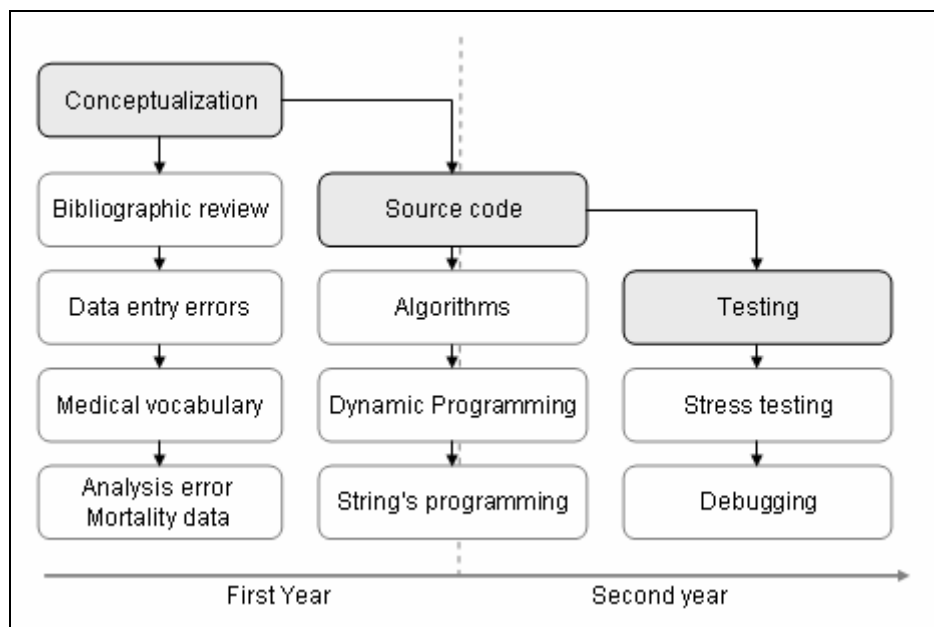
- Age:** A dropdown menu showing "0" and a text box containing "45" with the label "Horas" to its right.
- Sex:** A dropdown menu and a text box containing "Sex".
- Description:** A large empty text box with a "Search" button to its right.

## APPENDIX C

### Making ICODEX

The development of ICODEX is based in the following schema.

Figure C.1 – Making IcodeX, schema of the project



Schema about our design and methods, the process begins with the conceptualization, programming and the testing evaluation.

### C. 1 Conceptualization phase

This phase had duration of 1 year; it was focused on to determine the problems behind the manual coding for mortality data and the analysis of errors coding manually mortality data from Peru.

The work in this phase has two parts such as a bibliographic review of relevant research about problems (data entry errors, medical vocabulary) and an accurate analysis of errors for mortality data from Peru. The first part was based in solving two specific tasks, the systematic search and the systematic literature review.

#### 1 Search strategy

Simultaneous, but distinct literature searches were executed using on-line databases such as Pubmed, Medline, ACM Digital Library, NLM Gateway, CHID online, MathSciNet, MIT cognet, ANSI online, Electronic standards store and Engineering standards database search for six months. The search was completed with request for related research articles from EBSCO, ProQuest computing, Proquest medical library, NEC, and Elsevier-online Journals. The studies were located using the following list of keywords, sometimes as an alone word or a combined phrase.

1. Automated coding
2. ICD
3. Manual coding
4. Automated mortality
5. Coding Software
6. Controlled medical vocabulary
7. Mortality data
8. Coding errors
9. Health coding

## 2 Systematic reviews

- The systematic search was performed using standard methods described in Pubmed resources.
- The systematic review of relevant studies was based in our aims.

Our inclusion criteria were similar to the standard medical review which was published by the Cochrane collaboration resource. [Cochrane, 1996]

- Articles published in electronic journals from 1980 to 2003.
- Technical documents.
- We have also included articles published such as PhD or other research dissertations.
- Studies with statistical analysis and methodological criteria in accordance to Cochrane Collaboration methodologies.

## 3 Analysis of mortality data

The mortality analysis was performed following these three specific tasks:

- Collection of mortality data
- Depuration of errors using ICD-10 lists
- Descriptive analysis

We have used mortality data from the Ministry of Health from Peru from 1999 to 2000 for finding errors using ICD-10. The process of depuration was performed using extensive lists of validation and ICD-10 coding rules. In this sense, I was using the coding rules published in the volumes I, II and III of the ICD-10 books.

After the reliability analysis, I performed the descriptive analysis using the software SPSS for Windows version 11.00. This analysis was based in describing proportions and indicators of the following list of variables: Table X

Table C.1

List of variables used in the analysis for mortality data from Peru

<b>Variables</b>	<b>Type</b>	<b>Unity</b>
Place of death	Categorical	ZIP codes
Date of death	Censored	Date format
Age	Continuous	Numbers
Unity of age	Categorical	Five categories
Sex	Discrete	Two categories
Underlying cause of death (ICD-10 code)	Categorical	14,188 codes
External cause of death (ICD-10 code)	Categorical	14,188 codes
Physician certification	Discrete	Two categories
Hospitalized death	Categorical	Two categories
Accidental or violent cause of death	Categorical	Two categories

Source of the database: Ministry of Health from Peru, from 1999 to 2000.

ICD: International of diseases

ZIP: Geographic code from Peru.

Our outcomes were the percent of errors for each variable and the percent of errors for combinations of underlying cause of death with age and sex.

## **C. 2 Programming and testing phase**

Our scope in this phase was the development of a group of algorithms for coding ICD-10 for mortality data.

With all these concerns in mind, the following part of our work was the programming phase based in the main outcomes of our tool, which were organized in the following three main attributes. [Mc Call, 1977]

### Tool operation

- Correctness - How well the software performs the main functions and meets necessities of the health industry
- Reliability – How well the software can be expected to perform the coding process with required precision
- Integrity – How well accidental or intentional mistakes on the software can be withstood
- Usability – How easy it is to learn, operate prepare input of, and interpret output of the software
- Efficiency – Amount of computing resources are required by the software to perform its situation

### Tool revision

- Maintainability – How easy is to locate and fix an error in the software
- Flexibility – How easy it is to change the software

- Testability – How easy it is to say if the software performs its intended function

#### Tool transition

- Interoperability – How easy it is to integrate one system into another
- Reusability – How easy it is to use the software or its parts in other applications
- Portability – How easy it is to move the software from one platform to another

This phase was performed in two parts. The prototypes were primarily implemented in the LISP language programming using *Allegro Common Lisp for Windows version 6.0*. The final algorithms were written in Visual basic language programming using *Microsoft Visual Basic 6.0*. The translation was manually performed and it was necessary the use tables instead of text format for the ICD-10 codes. In this sense, I have also incorporated SQL sentences within the codes of programming.



## APPENDIX D

### The IcodeX system

IcodeX includes a database, the ICDL, and the IcodeX system. The IcodeX system is based on two forms and a module. The first form is the GUI of IcodeX which has the buttons and Text boxes for the sex, age and cause of death. The second form shows the list of possible candidates from the ICDL.

The module is based on eleven main functions. The Semi speller and the ICD analyzer are also based on these functions.

#### **DivideOracion(wTexto As String)**

Input: Strings of the cause of death

Process: Tokenization

Output: Validated strings of the cause of death

#### **Completa()**

Input: Validated strings of the cause of death

Process: Auto completion

Output: Complete phrase of the best possible cause of death

#### **BorraLetra()**

Input: Strings of the cause of death

Process: Delete erroneous characters

Output: Strings with similar prefix than the ICDL

#### **LlenaLista()**

Input: Validated strings of the cause of death

Process: Select 20 possible candidates from the ICD3

Output: Show the best candidates from the ICD3 for the requested cause of death

#### **LlenaFiltro()**

Input: Validated strings of the cause of death

Process: Filter the candidates for LlenaLista() and LlenaCIE()

Output: List of best candidates

#### **SeleccionaItem(TextoItem As String)**

Input: Outputs of the function LlenaLista()

Process: Select a candidate from the outputs of the function LlenaLista()

Output: Selected candidate from the ICD3

#### **LlenaCIE()**

Input: Selected candidate from the ICD3

Process: Select all the categories from the ICD4

Output: Show all of categories from the ICD4 for the selected candidate from the ICD3

#### **SeleccionaItemCompleto(TextoItemCompleto As String)**

Input: Outputs of the function LlenaCIE()

Process: Select a candidate from the outputs of the function LlenaCIE()

Output: Selected ICD-10 code

**LimpiaTodo()**

Process: Clean temporal variables and restart IcodeX

**LimpiaParte()**

Process: Clean text box of the cause of death

**GrabaEncontrado()**

Input: Sex, age and ICD-10 code

Process: record of inputs

## APPENDIX E

### The list of abbreviations

- ACME - Automated Classification of Medical Entities
- CDC-NCHS - Centers of Disease Control of the National Center for Health Statistics
- CLASSIC - Description logic based language
- COSTART - Coding Symbols for Thesaurus of Adverse Reaction Terms
- CPT – Codes of procedures and treatments
- DMT - Decision matrix table
- DOS – Disk Operating System
- DSM - Diagnostic and Statistical Manual of Mental Disorders
- GALEN - General Architecture for Languages, Encyclopedias and Nomenclatures in Medicine  
KL-ONE - Knowledge Representation Language
- GRAIL - GALEN Representation and Integration Language
- HCFA - Health Care Financing Administration
- ICD - International Statistical Classification of disease, injuries, and causes of death
- ICD-9-CM - International Classification of Diseases, 9th Revision, Clinical Modification
- ICD- 10-CM - International Classification of Diseases, 10<sup>th</sup> Revision, Clinical Modification
- ICD-10 - International Statistical Classification of disease, injuries, and causes of death, 10<sup>th</sup> Rev.
- ICD3 – ICD with three-digit codes
- ICD4 – ICD with four-digit codes
- MeSH - Medical Subject Headings
- ICDA - ICD Analyzer
- ICDI - ICD Information
- IC DL - ICD Lexicon
- ICDS – ICD Structure
- ICE - International Collaborative Effort on Automating Mortality Statistics
- ICPC - International Classification of Primary Care
- IC - Index of Categories
- INEI – National Institute of informatics and statistics from Peru
- ISBN - International Standard Book Number
- KRYPTON - Functional Approach to Knowledge Representation
- LOOM - knowledge representation language

- MED - Medical Entities Dictionary
- Medlee - Medical Language Extraction and Encoding
- MEDRA - Medical Dictionary for Drug Regulatory Affairs
- MeSH – Medical Subjects Heading
- MICAR - Mortality Medical Indexing, Classification, and Retrieval
- MIKADO – Coding systems from
- NCHS – National Center for Health Statistics
- PAHO - Pan American Health Organization
- SNOMED - Systematized Nomenclature of Human and Veterinary Medicine
- SRD - Statistical Research Division of the Census Bureau
- SuperMICAR -
- TRACER - Target Recognition of Automatically Coded Entity References
- TRANSAX - Translation of Axes
- UMLS - Unified Medical Language System
- UPC - The Universal Product Code
- WHO - World Health Organization
- WHOART - WHO Adverse Drug Reaction Terminology
- WHODRL - World Health Organization Drug dictionary