FRAMEWORK FOR CHARACTERIZING DATA AND IDENTIFYING ANOMALIES IN HEALTH CARE DATABASES

BY

ANNETTE M. SAVAGE

A THESIS

Presented to the Division of Medical Informatics and Outcomes Research and the Oregon Health Science University School of Medicine in partial fulfillment of the requirements for the degree of Master of Science
March 1999

SCHOOL OF MEDICINE

OREGON HEALTH SCIENCES UNIVERSITY

CERTIFICATE OF APPROVAL

This is to certify that the Masters thesis of

ANNETTE M. SAVAGE

has been approved

[redacted]

Karen B. Eden, Ph.D. (Professor in charge of thesis)

[redacted]

[redacted]

Paul Gorman, M.D. (Member)

[redacted]

Warren Harrison, Ph.D. (Member)

[redacted]

Associate Dean for Graduate Studies

TABLE OF CONTENTS

ACKNOWLEDGMENTS iii

Abstract iv

Introduction 1

Background 2

Describing Health Care Databases 2

Describing Anomalies Within Health Care Databases 3

Defining Anomalies Within Health Care Databases 4

Research Objective 7

Methods 7

Literature Review: Categorizing Bias 8

Developing the Initial Framework 12

Categorizing Processes 13

I. Errors 15

II. Missing data 17

III. Variation 21

Refining the Framework 25

Methodology: Validating the Framework 29

Methodology: Subjects 30

Methodology: Process 30

Results 32

Does the framework appear valid? 32

Can the framework be implemented as a physical database model? 32

Can the framework be used as a tool? 33

Table 1. Tabulated Results for Element Attributes 35

Table 2. Tabulated Results for Element Anomalies 36

Table 3. Tabulated Results for Process Step Attributes 37

Table 4. Tabulated Results for Process Step Anomalies 38

Discussion **38**

Does the framework appear valid? 39

Can the framework be implemented as a physical database model? 39

Can the framework be used as a tool? 41

Application of the Model 44

Conclusions **45**

References **46**

Figure Caption **49**

Figure 1 **50**

Figure 2 **51**

Figure 3 **52**

Figure 4 **53**

Figure 5 **54**

Figure 6 **55**

Appendicies ***56***

ACKNOWLEDGMENTS

My thanks to the many people who made this project possible: to my advisor Dr. Karen Eden for her direction and guidance; to Dr. Mark Helfand and Dr. Dale Kraemer for helping me refine my proposal; to Greg McCarty for assistance with the original meta data repository design, and in providing me with computer equipment; to Rick Wagner and Ben Chan for assistance with the outcomes claims data and allowing me to use them to informally validate the framework; to Larry Donohoe and Emily Campbel for technical assistance with Oracle as well as assistance with the outcomes claims meta data; to Li Zhang for assistance with Oracle and PERL; to Al Ferro for assistance with the UHC data; to Jeanne Ervin for her assistance with the prehospital care data; to Pat DeGarmo and Rick Brodner for their assistance with validation of the framework; to my fellow students and colleagues who gave me feedback on various aspects of my thesis; to my mother and Sherry Arasim for editing my thesis; and to the members of my committee, Dr. Christopher Dubay, Dr. Paul Gorman, and Dr. Warren Harrison, for their excellent guidance, support, and assistance.

ABSTRACT

As health care databases are becoming important sources of health care data, the anomalies these data may contain are of increasing concern. This paper addresses these anomalies by proposing a framework that can be used to characterize data in health care databases through meta data. This characterization of data is an aid to understanding the element structures and processes that influence and generate data in health care databases. The framework also aids in identifying and discovering anomalies in these data.

The framework was developed from past literature and research that identified anomalies and from an assessment of actual health care databases. An initial framework was developed from which a physical schema was designed. A meta data repository based on the framework was implemented. The implementation was used to refine the framework and to develop a new physical schema. The framework was validated on three aspects: 1) plausibility (Does the framework appear valid?); 2) technical feasibility (Can the framework be implemented as a physical database model?); and 3) feasibility of the framework as a model in terms of completeness and usefulness of the model (Can the framework be used as a tool?). The results indicate that the framework is a plausible model (Does the framework appear valid?) that can successfully be implemented as a physical meta data repository feasibility (Can the framework be implemented as a physical database model?). The result of validating the framework model indicates the framework is usable for characterizing processes and element attributes and for identifying anomalies (Can the framework be used as a tool?).

Since process attributes and element attributes produce and influence anomalies, certain types of attributes may serve to predict specific types of anomalies. Continued research in the refinement of the framework categories and the development of a more comprehensive

catalog of anomalies could then serve as a predictive tool for identifying anomalies and potential anomalies in a variety of health care databases.

INTRODUCTION

Health care databases are becoming important resources for health care data. These data are used for various analyses and research projects to help improve patient care while looking for ways of decreasing health care costs. Health care data document patient observations, treatments and services provided, and assist clinicians in addressing the needs of individual patients [1]. Health care data are used to improve treatment effectiveness and to evaluate the variability of health care delivery and its associated effect on patient outcomes such as costs, morbid complications and death [2, 3]. These data are used for planning and policymaking by administrators and for meeting the demands of managed care such as monitoring utilization of services and supplies [1, 4]. Health care databases provide access to large amounts of medical data. However, these databases can also contain problems such as errors, missing data, inconsistencies, and misrepresentative data. These problems are termed "anomalies" in this paper. An *anomaly* is anything that is or seems to be inconsistent, contradictory, or improper [5].

Health care databases may also have such differences between them that reconciling these differences may create additional *anomalies* in the resultant database. Anomalies can create difficulty when using these data. Anomalies can skew or bias the data. Anomalies may also cause the data to inaccurately represent the original observation, measurement, or intent of the data. Anomalies may adversely affect decisions, analyses and inferences made from these data.

Although one may wonder why some anomalies exist or why they cannot be kept from occurring, it is important to understand that some values that are valid for their primary purpose, may be anomalous values in subsequent uses. For example, in one database that was assessed blood pressure was recorded as two separate fields (systolic and diastolic).

Because the measurements were often taken in the field under less than ideal conditions, the software was designed to allow for the diastolic measure to be either a number or the letter "P" designating the blood pressure was palpated. Although appropriate for the primary use, any use of diastolic pressure as a number required the conversion of the field to numeric and the removal of all anomalous non-numeric values. Because of this difference in how data are used, the characterization of these data becomes very important and is a key component of this research.

BACKGROUND

DESCRIBING HEALTH CARE DATABASES

Health care databases are used for a variety of purposes. They can be stand-alone databases used for specific research or exist as a part of a larger hospital information system. Types of health care databases include: population-based databases, such as census data and vital records; compiled databases, such as ad hoc survey and research databases prepared by investigators in academic, government, and private organizations; hospital discharge abstracts; disease registries; computerized medical insurance claims data; computerized health care administrative data and fiscal data; and databases supporting electronic medical records or clinically rich databases [6, 7]. Laboratory and other ancillary databases also provide important health care data.

These databases contain source data for a large number of people and are often representative of a general population or large subsets of the population. They are used to profile a patient population and the accrual of patients over time [8]. Health care databases are used to determine sampling frames, a means for identifying cases, and to estimate the total number of cases in the population of interest. These patient populations are then contacted for subsequent primary data collection, for example, surveying all diabetic patients on health status [6]. Health care databases may be used for clinical research especially to help in the generation of hypotheses [6, 8, 9]. Health care databases may provide an option for representing experiments where it would be unethical or impractical to attempt to randomize subjects in clinical trials into treatment and control groups [2]. Health care databases are compiled from different locations and used to obtain data in cases of rare diseases or for unusual procedures [8].

Medical databases, a type of health care database containing clinical observational data, have been important for developing clinical prediction rules and help confirm or extend findings from other studies [9]. Population-based databases provide access to incidence, prevalence and mortality rates of disease [8]. Large observational patient databases provide data important for determining optimal patterns of care and ideal treatment strategies [7].

DESCRIBING ANOMALIES WITHIN HEALTH CARE DATABASES

Health care databases are any collection of health care data whether existing as records in patient files or records in an electronic database. Both paper and electronic records suffer from similar anomalies, especially when using these data for research. Although existing health care databases are used for various research projects, they may not have been designed for research purposes. For example, the process of acquiring data in many health care databases does not necessarily follow any specific research protocols for the collection of the data, which may affect the reliability of the data [1, 10]. Even so, many people still use these "non-research" type databases for research.

It would be difficult to design and compile health care databases to meet the needs of all users, and as a result there are several types of databases to meet different needs. Because of this, the data needed for a particular analysis or research effort may not exist in the database [6]. For example, in order to control for confounding factors adequately, data about these factors and how they were collected will need to exist [8, 11]. Furthermore, it may not be clear which variables in the database are unreliable, especially to someone not involved in the data-gathering and recording process [9]. In addition, treatment assignment for patients in non-research databases is usually a result of patient prognosis and physician preference, and as a result, data in health care databases may contain selection bias due to the non-random assignment of treatments [8]. This will affect inferences regarding treatment efficacy that can be drawn from these data. Even when data in health care databases seems complete and correct, it may not accurately reflect the status of the patient because what is recorded about a patient is not always what health care providers know about a patient [1].

Although paper and electronic health care records share similar anomalies, the focus of this research is primarily with electronic health care databases. For researchers, electronic health care databases provide existing sources of health care data that saves time and money for data that have already been collected by others [2]. However, electronic health care databases are often not compatible with other health care databases in that they may contain different definitions, and coding conventions as well as file structures which may create anomalies when trying to merge and analyze these data together [2, 4]. The complexity of hospital information systems, from which many of these health care databases come, combined with a variety of developers and poorly developed and/or adhered to standards for data representation make combining these databases a difficult task [12-14]. Furthermore, codes and even record layouts change over time affecting any analyses which would span the time period across changes [6, 11].

DEFINING ANOMALIES WITHIN HEALTH CARE DATABASES

Anomalies occurring in health care databases can be categorized as those involving invalid values; values that are valid but misrepresent the true observation, measurement or intent of the data; inconsistencies; and missing values [11]. Invalid values may be a random occurrence such as a typographical error, or may not be random such as systematic errors. This research is concerned with errors that are not random. Ideally, systematic errors can be prevented or reduced. For example, software data entry programs that require the user to enter a value into a field, whether an entry would be applicable or not, may find that a value has been entered merely to satisfy the requirements of the program rather than to record valid data.

Data may be misrepresentative in a number of ways. For example, some systems restrict the possible values that can be entered to a pre-determined list. In this situation, if the correct value or description cannot be found, a value may be entered that seems to be the closest, but does not truly represent the real situation or observation [6]. Misrepresentation may become more pronounced when trying to merge databases. Attempting to create a common set of codes from different sets of codes assumes that a common process for coding and understanding code definitions was used. However, conventions for coding may vary and change over time and as a result, merged databases may not truly represent the original data or patients [2, 6].

A data value may be valid and accurately represent the patient condition or observation, however, when considered with other values there are inconsistencies. For example, if the definitions of codes are unclear, one person may use one code to describe a situation and another person may use a different code to describe the same situation [6]. This will not only create skewed results when trying to analyze the data, but may be difficult to discover. Inconsistencies may also occur when the value for one or many data elements is not appropriate based on values for other data elements. For example, a procedure date is prior to the admit date, which may or may not be valid depending on the organization or situation, or a procedure is invalid for the age or sex of a patient [15].

Missing data may be due to a variety of reasons, for example, the value was not observed; it was forgotten; or it did not seem necessary to record [16]. It is difficult to know if a value is missing because it was not assessed or was not documented [1, 11]. It is important to note that missing data, although perhaps a problem for the analyst, may be acceptable for those making the observations and recording the data [16]. In addition, entry of a value may be related to factors such as time pressures and the perceived importance of the item [17].

Data elements may suffer from one or a combination of these anomalies making the data more unreliable and analysis more difficult. Ideally, preventing anomalies in the first place would improve data in health care databases. This may involve interventions such as requiring a strict adherence to standards [10] (using the same vocabulary and standardizing measurements), exhaustive data entry checking, structuring all data, etc. Unfortunately these attempts may create different anomalies. For example, restricting observations to a list of coded elements may prevent a health care provider from accurately describing a patient's condition. Attempts have also been made to correct the data after it is collected. For example, missing data values may be determined using statistical approaches and artificial intelligence [16, 18], but there is no guarantee this process will not create additional anomalies. It would be unrealistic to assume all anomalies could be prevented or corrected. Correcting anomalies in the data may result in more anomalies. Regardless, it is important to know what anomalies could exist and to identify their source.

Anomalies in health care databases may or may not be apparent thus a method to discover anomalies is useful. In order to discover the anomalies in health care databases, an understanding of the characteristics of these data is required.

RESEARCH OBJECTIVE

This project proposes characterizing data in health care databases with meta data, or information about the data, as a way to understanding these data and identifying and discovering anomalies. As stated earlier, an anomaly is anything that is or seems to be inconsistent, contradictory, or improper [5]. An anomaly may be dependent on the perspective of the person using the data. For example, normal values are not recorded in a clinic, which may be appropriate for clinic use of the data, but may cause anomalies when attempting to analyze data in research [19].

The purpose of this research is to develop a framework that characterizes data in databases by classifying data element attributes and process attributes, and by classifying anomalies. The framework could eventually be used as a tool for identifying anomalies in databases.

METHODS

The development of the framework began with a review of literature to discover the types of anomalies that have already been found in data from health care databases. From this information, a list of categories describing areas where anomalies may occur in data was developed and used as a backdrop for the development of a framework. Working backward from a description of possible anomalies, an attempt was made to discover real anomalies in actual databases and in doing so discover what information would need to be known about the process and element attributes that produce and influence anomalies. It is hoped that by understanding what process and element attributes produce and influence anomalies that it will be easier to identify anomalies in other databases. With each database assessed, the framework was refined and used for the next iteration.

LITERATURE REVIEW: CATEGORIZING BIAS

The original goal of developing the framework was to develop a taxonomy or classification of biases. The idea of categorizing bias in health care databases came from an article by Robert Dolin (1997) which discussed categorizing types of bias which may occur in electronic medical records [7]. Dolin used the term bias traditionally as that which can affect the validity of a study by either causing a study to show a statistically significant result when there wasn't one or by showing no difference when there was a significant difference [7]. The traditional statistical definition of bias seemed a good starting point for beginning to describe anomalies and the effect anomalies could have on data in health care databases. However, the traditional definition of bias seemed confining in that there may be anomalies in health care databases that are of concern, but do not necessarily lead to traditional statistical bias. Bias was expanded to include any source of variation that distorts data whether it led to statistical bias or not. Using this definition an expanded list of six possible types of bias was constructed. To help illustrate different bias types, examples from David Sackett's (1979) catalog of biases are included [19].

1. INTERVENTION BIAS. Biased intervention includes the differences that can occur in the scope and degree of intervention performed on a patient [7]. Unlike clinical trials, decisions for treatment and intervention may be based on patient prognosis and physician preference, or other undefined parameters [7, 8]. An example of intervention bias could be:

   Procedure selection bias. Certain clinical procedures may be preferentially offered to those who are poor risks. [19]

2. SELECTION BIAS. In addition, intervention bias may occur simply as a result of the types of patients seen at the clinics where the data were captured resulting in selection bias [8].

With selection bias, one cannot be certain that the population being studied in a database is the same as the population in which a researcher is interested [7]. Most likely, selection bias cannot be corrected, but documenting the patient base from which the data were drawn may help alleviate the effects of this bias [8]. Documentation may include the rules for inclusion or exclusion of records in the database [6]. Examples may include:

> *Popularity bias.* The admission of patients to some practices, institutions or procedures…is influenced by the interest stirred up by the presenting condition and its possible causes.
>
> *Centripetal bias.* The reputations of certain clinicians and institutions cause individuals with specific disorders or exposures to gravitate toward them. [19]

3. DEFINITION BIAS. Biased definition covers a variety of concerns about the meaning of data elements. The meaning of data elements is complicated by the lack of a standard nomenclature for describing medical data [7, 12]. Even if a standard vocabulary is identified for a clinical system, there may be differences in consistency of how those definitions are applied [8]. For example, individuals may code the same observation using different terminology [6]. In addition, concepts and meanings of codes and terms may change over time [6, 7, 11, 12]. Sasse (1998) describes problems with definitions as consisting of aliases, imposters, and hybrids. Aliases occur when more than one data element is used for the same purpose. Impostors arise when more than one data element has the same name, but they have different operational definitions. Hybrid data develop when one data element has more than one operational meaning or use [20]. An example from Sackett's catalog:

> *Diagnostic vogue bias.* The same illness may receive different diagnostic labels at

different points in space or time, e.g. British 'bronchitis' versus North American 'emphysema'. [19]

4. MEASUREMENT BIAS. Biased measurement describes errors made while acquiring measurements or making observations [7]. This includes the differences between observers in the methods used to obtain measurements and physical findings [7, 21]. Measurement bias also includes the possibility that test accuracy and reliability may change over time. In addition, this bias includes the differences between instruments and the different ways used to measure the same variable [6, 7]. A specific example might be:

   Instrument bias. Defects in the calibration or maintenance of measurement instruments may lead to systematic deviations from true values.

   End-digit preference bias. In converting analog to digital data, observers may record some terminal digits with an unusual frequency, e.g. a notorious problem in the measurement of blood pressure. [19]

   Measurement and definition bias may be corrected by working with those who record the data to ensure that consistent recording methods and vocabulary are used and that data are recorded as objectively as possible [9]. However, these interventions may be after-the-fact, and data collected before this intervention will need to be scrutinized for inconsistencies. Arndt et al. (1994) discovered mistakes are often made at the time of measurement and the number of mistakes was often related to differences in the perceived importance of a data element and how closely the element related to one's field of study [17].

5. RECORDING BIAS. Recording bias includes what was not recorded, or missing or incomplete data [6, 7, 16, 18, 21]; and errors in recorded data such as transposed or

extreme values [21]. Although the focus of this paper is on systematic data entry errors, recording bias includes both random and systematic errors both which may introduce erroneous data. Some of these data, considered contaminants of the database, cannot be accounted for or removed since manual inspection of large databases may be impractical. An understanding of this limitation is important when using these data for research [22, 23]. Examples include:

Missing clinical data bias. Missing clinical data may be missing because they are normal, negative, never measured, or measured but never recorded. [19]

6. STRUCTURAL BIAS. Structural bias recognizes the differences in how data are collected, such as whether data are collected through structured entry or free text entry. Free text entry is inherently more difficult to extract knowledge from, and may prove costly as well as limit the effective use of the data [12, 18]. Prather et al. (1997) determined that free text stored in place of coded data accounted for the largest amount of unusable data [18]. In addition, current technology is still unable to interpret and extract usable information from narrative or free-text entries [24].

Furthermore, structural bias can occur when comparable data values differ in granularity [6, 13]. Granularity is the level of detail in which a data element has been recorded. For example, time may be recorded in seconds rather than in minutes. Some differences in granularity can be corrected by converting the value from the more granular entry to the value that is less granular [13]. Caution should be taken as this procedure may result in truncating or combining some values in favor of a more general entry and possibly altering the intended meaning of the recording.

Structural bias is often a function of the entry interface software, through which the user enters the data. Correcting or altering the software is not often possible, so an awareness of this limitation is important. For example, systems often require that blood pressure be recorded as two numbers, a systolic and a diastolic pressure. However, in certain patient care environments, such as in emergency field response, the blood pressure is often palpated and there is no diastolic recording. Unless the system allows for recording that the blood pressure was palpated, the system may have a number of missing diastolic values with some uncertainty as to whether the recording was missing because the blood pressure was palpated or for some other reason.

DEVELOPING THE INITIAL FRAMEWORK

The first database used to help develop the framework was not actually a database but data analysis reports from the University HealthSystem Consortium (UHC) for Oregon Health Sciences University data. Briefly the UHC is an alliance of academic health centers, which compiles a clinical database that provides comparative clinical, administrative, and financial data. These data are used by the member academic health centers to provide comparisons on resource utilization, charges, costs, and length of stay. The reports describe actual anomalies found in the UHC database.

The UHC data analysis reports provide several types of information. Among them are reports for each dataset that identify the numbers of valid data, outlier data, and bad data for each member health center. They also provide a report that identifies items they feel would be of concern to data analysts such as centers that have updated their data from previous releases; differences in business practices among the centers that may affect analysis; accounting for DRG code changes; etc. They also provide a detailed listing to each center analyzing that center's data, summarizing anomalies and flagging records that seem to contain errors, inconsistencies, or missing data [15].

In addition to the UHC reports, another database was used to refine the framework.

This database was compiled from two source databases containing claims data for OHSU being merged as part of a data warehouse project. The exercise of merging databases was assessed, however, there was minimal information on the processes used to acquire the original data. There was also limited access to the actual data and discovering anomalies within the database was difficult.

To formally identify anomalies in health care databases, it is important to understand the processes that the data in these databases go through both before and after the data are stored in a particular database, on a report or in a file. At each step of a process there are points at which anomalies may occur. In addition, anomalies may manifest only within a particular context (e.g. attempting to identify all diabetic patients in a database or when comparing two or more databases with each other). What follows is a detailed classification of the process and possible causes of anomalies.

CATEGORIZING PROCESSES

In trying to understand the processes used to acquire, transform, and analyze data, an attempt was made to list steps data may go through from the point of acquisition by a human or device to the point of use by an analyst (see Figure 1). As seen by the author, the data may go through several types of processes from the point of acquisition to use (e.g. recording, transcription, updating, correction, merging, extraction, etc.). They can be grouped into three major types, which are listed below with examples:

1) **ACQUISITION** – Any process involved in acquiring, generating, and storing data. Acquisition may be by either human or device.

   RECORDING MEASUREMENT OR OBSERVATION – Includes entry onto a paper chart, dictation for transcription, or electronic system. Entries may be either free-text or structured.

TRANSCRIPTION/CODING – Includes taking data previously recorded in one form and transcribing or re-coding into a database.

UPDATING/EDITING – Includes the rules and edit checks applied to the data either manually, by device, or software at the point of entry or before the entry is considered "final" or saved.

2) **TRANSFORMATION** – Any process which transforms the data from a previous database to another database or form.

EXTRACTION – Includes extraction to a manual or computerized form or report. This may include transformation into another database system such as to a data warehouse.

MERGING – Any process which merges two or more databases (or subsets of databases – data sets) into a common database (e. g. data warehouse).

3) **ANALYSIS** – Any process that uses data once it is stored in any database.

In view of this design, it would seem the process in which the greatest number of potential anomalies may occur would be the acquisition process. Once the data are in permanent storage, changes to the data are less likely to occur.

The purpose of identifying the processes used to acquire, transform, and analyze data was an understanding of what led up to the development of anomalies. However, at first, the end result was to focus on the anomalies and somehow categorize all that could be wrong with the data. The initial framework categorized potential types of anomalies. It was broken down into anomalies caused by errors in the data, anomalies caused by missing data, and anomalies resulting from attempting to merge multiple databases or *variation* between databases or between subsets of a single database. It would have been an enormous task to attempt to categorize all anomalies that could potentially occur in all databases. The exercise of discovering anomalies and cataloging them all would have been impractical in the time needed to complete development of this framework, and perhaps may never completely be known. So the option was left open to add categories as needed. The initial framework (see Figure 2) organized anomalies by process type (acquisition, transformation, analysis) and types of anomalies (errors, missing data, variation between databases).

A description of the types of anomalies shown in the initial framework is detailed along with descriptions of each category. The descriptions start with discussing errors and the occurrence of errors in the acquisition process, then the transformation process, and finally the analysis process. All missing anomalies in each process type will be discussed and finally all variation anomalies in each process type will be discussed.

I. ERRORS

Although errors may easily occur as random errors, they may also occur as the misapplication of the rules designed to ensure data integrity. Examples are selecting the wrong value from a list of allowable values for a field, selecting the incorrect value to use in a given situation, or violating the rules for ensuring data consistency.

A. ACQUISITION ERRORS

DATA ENTRY – This includes such errors as transposition errors of numbers, misspellings, typographical errors, etc. These are more or less the random errors that may occur in collecting and recording data. These errors are non-systematic and so are not included in the framework.

INVALID VALUE – Any single element value that is invalid. This may include a value that is out of range, lacks the correct number of significant digits for a valid identifier (such as for a social security number) or a value that does not match the data type (such as entering a character where a numeric should be entered) etc. Some of these may seem to be random errors such as lacking the correct number of significant digits or entering characters where numbers should be. However, investigation into databases showed that in fields where values are required, occasionally people purposefully entered incorrect entries simply to satisfy the demands of the program. Unfortunately, separating which errors are random and which are systematic may be difficult in some cases. It may seem that invalid values should be eliminated through programming the criteria for validation into the interface or database. However, this is not always the case perhaps due to attempting to provide flexibility for allowing occasionally seemingly invalid values that are on occasion valid, or for allowing for differences in the expression of medical events, etc.

MISREPRESENTATION – These are values that are valid, but are the wrong value for the situation. For example, as mentioned earlier, systems that restrict the recording of an observation to a coded list may risk misrepresentation if the correct code cannot be found such that a code that seems closest to the correct code is entered, yet it does not accurately describe the observation [6].

CONSISTENCY – These errors include the violation of some criteria that exists for a set of elements usually based on some condition or set of conditions. For example, a patient is recorded as being pregnant while the sex of the patient is recorded as male. These types of errors often occur systematically when default values are used and the user doesn't change the default. Another example is if the same information can be recorded into more than one location and there is a difference in agreement between the two recordings.

B. TRANSFORMATION ERRORS

TRANSFORMATION ALGORITHMS – Transformation errors may occur when transformation algorithms are applied to data in an attempt to conform data to standards and formats different than the original format and values. The result could be one of four outcomes: valid data becomes invalid; invalid data stays invalid; invalid data becomes valid; valid data stays valid. The real problem may be determining which one of these outcomes has occurred. If this can not be determined, the reliability of the data is compromised.

Data may also become inconsistent or may no longer accurately represent the original situation. For example, attempting to conform multiple code sets into one standard code set may result in forcing multiple codes into a single standard code, or in a situation where a standard code has no equivalent in a source code set. In the first situation, a value may be over-represented whereas in the second situation a value may be under-represented.

C. ANALYSIS ERRORS

WRONG ASSUMPTIONS – Analysis errors, in terms of the data, may result from misinterpretation due to assumptions made about the data that are not correct. For example: the assumption that a value is an error when it is not an error, or the assumption that a value is correct when it is not correct. Assumptions may also be made about the representation, consistency, and reliability of the data.

II. MISSING DATA

A. ACQUISITION MISSING DATA

There are several causes or reasons for missing data [16]. It is important to note that many of the reasons are appropriate, such as absence of data may indicate a normal value or recording a value would not be applicable to the situation. A breakdown of possible reasons, observed by the author and collected or determined from research is discussed in the following paragraphs. Although time is a consideration for all potential anomalies, it seems that it may play a significant part in promoting missing data. The less time a clinician has to collect and record data, the greater potential there is for these data to be missing [16].

NOT APPLICABLE – Entering a value into a specified field is not necessary for the particular situation. For example, if a patient wasn't transported to the hospital, then fields requesting data about the transport are not applicable.

UNKNOWN – The value was not known at the point when the data were being entered into the database. It could be the data were never known or lost at some point in the acquisition process.

NOT MEASURED – The data element was never obtained. If the person gathering the original data is unaware there is a need for a particular data element, then it may be less likely the data will be obtained in the first place [19].

NOT NOTEWORTHY – The person recording the data did not feel the event or observation was noteworthy enough to record.

NOT NECESSARY – The person believes the most important data are already present and so other data could be omitted [16, 19].

NOT RECORDED– The data were overlooked or forgotten when entered into the database [16]. In addition, some information may be sensitive in such way that the patient or clinician does not wish the information to be recorded [25].

MIS-RECORDED– The value was entered into the wrong or another place. In a systematic way this may occur when users are not welled trained or software allows entering the same observation in more than one place. For example, software that provides pre-defined lists from which users record observations may also provide another free-text field in which users may enter observations not on the list. In one database that was assessed, some users used the other free-text field to record items regardless of whether they were on the pre-defined list or not.

LOST DATA – The data were lost at some point in the acquisition phase or while in the database. This could be from system crashes, program bugs, or data maintenance errors resulting in the inadvertent deletion of data. This may sometimes occur unexpectedly during upgrades [1, 16].

NORMAL VALUES – Values representing a normal situation or condition are not recorded [19].

B. TRANSFORMATION MISSING DATA

Missing data for transformation focuses on transforming only subsets of data. Some of this transformation is for the purpose of "cleaning" the data in an attempt to make the data more useful for a specific purpose. Some examples include:

EXCLUSION OF ROWS AND /OR RECORDS – Selecting only a subset of rows for transformation based on some criteria (e.g. all Medicare patients for inclusion on Medicare forms).

EXCLUSION OF FIELDS – Not including fields in a transformation based on some criteria. For example, excluding fields with a high amount of missing values, or assuming fields are not important or necessary for inclusion in the new data set.

EXCLUSION OF VALUES – This is the exclusion of a value in a field based on some transformation rule. A field that allows characters may have the values containing characters removed during transformation so the remaining numeric values can be stored in a numeric type field. For example, in a pre-hospital care database diastolic blood pressure allowed the user to type in a numeric value or the letter "P" to designate that the blood pressure was palpated. If someone wanted to analyze the diastolic pressure as a numeric value, the field would need to be converted to a numeric type and the palpation designation would need to be recorded some other way. Another example can often be found in research databases where users enter values such as 99:99 for an unknown time which is also an invalid time, but these databases are constructed to hold these values because this method of coding is a standard. In addition, some values may be excluded for confidentiality reasons such as values that can be used to identify a patient as in social security number, or location of where a patient injury took place as in an automobile accident. These may occur as part of the "data cleansing" process in a data warehouse project.

C. Analysis Missing Data

Anomalies caused by missing data in the analysis phase is often a function of the question or questions being answered and the data needed to answer those questions such as only studying diabetic patients. In this way, there are various reasons why only subsets of the data may be used.

Eliminating Rows/Records – For example, eliminating rows with missing data, or because they do not fit some criteria. Analysis may also attempt to account for data that is assumed to be in error by deleting those rows containing values thought to contain bad or unreliable data or outliers.

Eliminating Fields – All the fields may not be necessary for analysis, and in this way fields may be eliminated from the final analysis. A concern is if remaining fields were dependent on the field that was eliminated [6]. For example, there may be several fields that determine a patient is a diabetic patient such as diagnosis codes, lab results, and pharmaceuticals. Eliminating one of those fields may make it difficult to determine which patients are in fact diabetic.

Eliminating Values – Analysis may also attempt to account for data that is assumed to be in error or which may cause anomalies by deleting or excluding those values thought to be in error such as outliers.

After the initial framework was developed, it became apparent that although analysts were involved in processes that eliminated rows, fields, and values they were still performing transformation functions. This was one of the reasons the initial framework was later revised.

III. VARIATION

A. Acquisition Variation

Variation in the acquisition process may exist at many levels from methods used to collect data to the structure of the data themselves. Variation may also exist in the events and decisions used to determine how patients are treated. Although these events affect the data, it is not the purpose of this framework to assess these events for their influence on variation in the data. The purpose of the framework is help those using data in health care databases to know that any difference they may find in the data is real and not a result of some error or variation in the data themselves.

There may be many ways to categorize variation. A list of variations is included in the sections that follow. Although it may not be exhaustive, the list seems to represent variations that may be of concern. It may be important to note that variation usually only exists when comparing more than one database or data set. However, there are instances where variation may exist within a single database or data set. Variation within a database or data set may be due to a normal progression of change to reflect changes in the organization, practices or guidelines. Variation within a database may also reflect the inconsistent application of rules to the data which may result in compromising the data integrity [26].

Structural – Structural variations are differences in how data are structured when entered into permanent storage. For example, this may include differences between structured entry, where data are broken down into defined fields, and free-text entry, where those recording the information are free to select their own values and/or define their own structure. This includes how dates are formatted, as well as numeric fields, etc. Differences in how data are organized and stored over time are also included. For example, systems that keep audit trails, or addenda for recording changes to records, may be keeping important information in separate files or tables. Some of these data may be redundant and some may not. A system that is structured such that it is possible to store a single data value in more than one location, field or form, may also promote anomalies when attempting to analyzing these data. A final example for structural variation is the level of detail or granularity recorded about an item.

Definition – Definition variation really includes differences in what the definition of a certain element is assumed to be by those recording the data. These definitions are more commonly referred as operational definitions. It's important to note how a software vendor defines what data should be entered into any particular field may not be what is in fact entered into that field. Operational definitions may also differ between individuals collecting data. For example, different individuals may have different criteria for deciding what severity level to assign to a patient.

Method/Measurement – Method variation covers a wide area because the methods by which data are collected, recorded, edited and stored may vary greatly.

These differences may include human versus machine collection of data, or differences in how measurements are made. For example, if pulse is always counted for 6 seconds then multiplied by 10, the resulting value will always be a factor of 10 creating a stair step graph of pulse rates rather than a smoother line more reflective of actual values.

CONSTRAINTS/RULES – Constraint variations can best be defined as those rules that are applied to data either through human enforcement or software enforcement in order to restrict or constrain what values may be entered into a field. This may include criteria used to decide what values to enter such as standard coding practices, to relationships which are enforced within the data, to set default values, etc. Limits or ranges for certain data values would be an example of a constraint. The set of codes from which an individual may select would be another. Default values may be of special concern since it may be quite easy for someone to accept a default value. In this case knowing what fields have defaults, what the default values are, and then knowing how often the default occurs may be very important to analysts.

TEMPORAL/HISTORICAL – Temporal variation really refers to the changes that may occur within a single database or system over time. The changes cause a difference between the data collected before the change and the data collected after the change. For example, the UHC (University HealthSystem Consortium) database, with compiled data from several university hospitals, reported that OHSU's submitted database had used codes that no longer existed and were now invalid when compared with the new set of codes. UHC also states that in order to locate an appropriate DRG name in the DRG name file, it is necessary to match by DRG and DRG version. This is because the same DRG number can have a different definition and description for different discharge dates. This is another example of variation over time.

RECORD KEEPING PRACTICES – Some record keeping practices may affect how the data can be used during analysis. Consider the following: two hospitals (in the UHC analysis report) report room and care charges for some newborns on the mother's record assuming when these babies are healthy, they will have very low charges per day. The UHC report also states three hospitals do not provide ICD-9-CM codes for a number of diagnostic procedures that are performed at their facilities and that one hospital does not provide ICD-9-CM codes for a number of non-billable procedures.

BUSINESS PRACTICES – Some business practices will affect what data are available for analysis. For example, from the UHC analysis report, eight hospitals do not provide services to maternity cases and one hospital did not provide services to maternity cases prior to January 1, 1996.

PURPOSE – The reason for which data were originally collected gives one an understanding of the type of data elements likely to exist and perhaps alerts one to the limitations on the values recorded [1]. For example, claims data may contain information on medications given to the patient, but those entering medications may only be concerned with medications for which they can get reimbursement. Understanding that a claims database supports the reimbursement process may bring to light types of data that may or may not exist in the database. In addition, data in health care databases have not been collected with a specific research purpose, and this will affect the usefulness of these data in research [11].

B. Transformation Variation

How Data are Transformed – Variation occurring during the transformation process may be produced by the differences in how data are transformed. Differences include: changes which may occur as a result of altering the structure or granularity of data, or ways in which differences are reconciled such as merging different code sets or coding systems [6]. These errors may occur as part of the "data cleansing" process as in a data warehouse project.

C. Analysis Variation

How Data are Used – Variation in analysis probably centers on how the data are used and what questions are being asked of the data. The framework does not expand on this as the variations can be many, though it may be important to note differences in what the data were collected for and how they are being used.

Assumptions – Since there may be assumptions being made about the data in any analysis effort, it may be worth recognizing and stating them. For example, an assumption that all medications are entered into the system when in fact only billable medications are entered.

REFINING THE FRAMEWORK

Using the initial framework (Figure 2), an additional database was evaluated. This database was a clinically rich database containing information on patients involved in emergency prehospital care incidents. The acquisition processes were easily available for this database as well as access to the data.

In this iteration of the framework information came to light about the data that could not be categorized as a known anomaly, but may be important for understanding the data and for discovering potential anomalies. For example, in the prehospital care database each patient record has a unique patient identifier. However, in prehospital care a unique patient identifier is only unique for a single patient involved in a single incident, an incident being any event to which emergency personnel respond. If the patient is involved in another incident, then he or she is assigned another unique patient identifier. If these data are ever merged with other patient databases, this information would be important to know, yet in itself it is not an anomaly.

Most descriptions of anomalies that were acquired by analysts were not only a description of the anomaly, but also a description of the part of the process or element attributes contributing to the anomaly. For example, in describing a past anomaly in the prehospital care database regarding IV (intravenous) attempts, analysts also described the solution to this anomaly and identified when the fix was made. The anomaly was still important to record as it still applied to all data recorded prior to the correction, but the new element attributes (structure changes in the elements) also seemed important for understanding the data recorded after the correction was implemented. In the past, IV attempts were designated by a count, then responders were asked to record whether there was a successful IV attempt. It was discovered that responders varied in how they interpreted this data requirement. Either they recorded the total number of attempts including the successful attempt in the count, or only recorded the count of unsuccessful attempts. Altering the element attributes and simply asking responders to record the count of successful and the count of unsuccessful IV attempts later fixed this anomaly. This change in the IV element attributes was important information to record, yet the framework only focused on the anomalies and so attribute information such as this had no place to be recorded.

It was apparent that only focusing on the anomalies in the database was not enough, but rather a characterization of the element attributes and process attributes in these databases was important in order to understand how anomalies developed or could potentially develop. For example, although not an anomaly, the process involving how response times in the prehospital care database were acquired greatly affects how the data will need to be retrieved and used later. Responders record response times by pressing the appropriate status button in the response apparatus, which sends a signal to the dispatch computer to record a time associated with the type of button pressed (responding, arrival, etc.). Sometimes responders were not sure if the correct button was pressed so they would press a single response type button (responding time, arrival time, etc.) multiple times. This resulted in multiple response times for the same response type being recorded. When using these data, it is assumed the first response time is the correct time. Again this is not necessarily a specific anomaly description, but a description of process that significantly impacts how the data will be used.

The framework was refined, focusing on process and element attributes and their role in the development of anomalies, by providing areas where element and process information could be recorded. In this model the relationships between the elements and the processes that created them are key. The resulting framework is shown in Figure 3. This is a generalized diagram showing the relationships between processes and elements and the relationships between process attributes that produce anomalies and elements that suffer from anomalies. The shadowed attribute boxes are holders showing where items formerly listed under the initial framework would exist in the refined framework.

The items from the initial framework that were moved into the attribute boxes were those items listed under the category "Variation." Variation was originally created to account for differences between databases, which may cause anomalies when comparing databases or data sets together. These categories seemed appropriate not just for finding anomalies between databases, but for simply characterizing processes and elements.

Those categories that specifically characterized elements were listed under element attributes. This included structural attributes, element definitions, and information on relationships that a data element may have with other data elements. For example, if the existence of value for a data element is dependent on a value in another data element, then the dependency is noted in the element attributes. As an example, in the prehospital care database if the trauma system entry field contained the value "Y" for yes, then a value in the trauma band id field would exist, otherwise a value would not exist in this field.

Those categories that specifically characterized processes were first organized under six questions (see Figure 4 which shows a complete list of the attributes and anomaly types, or Appendix B): 1) What is the process? 2) Who were involved in the process? 3) When did the process take place (e.g. day and time of day)? 4) Where did the process take place? 5) Why was the process performed or initiated (what was the purpose)? 6) How was process carried out? The under the "How" question are listed the other non-element categories formally under "Variation" in the initial framework. Under the general process phase of acquisition the categories include: business practices; measurement and observation methods; record keeping practices; constraints and rules applied; and changes in any of these methods over time. Under the general process phase of transformation categories include: extraction rules; mapping rules; scrubbing algorithms; and inclusion and exclusion criteria. These categories although seemingly able to account for all attributes encountered in this study, are not necessarily a final list. In order to account for not yet identified categories or variation in grouping attributes, this framework remains "unbounded" allowing for categories to be added.

Figure 4 shows a complete list of the attributes and anomaly types. This model describes the structure of meta data needed to describe process and element attributes as well as anomalies. The unbounded nature of the framework is designated by the "other" category listings.

A more formal model can be viewed in Figure 5. Figure 5 is a detailed diagram showing the relationships and entities necessary for formally implementing the framework as a series of tables in a relational database repository and discussed in the results.

METHODOLOGY: VALIDATING THE FRAMEWORK

The validation of the framework is a prerequisite to a formal evaluation. The validation shows the reasonableness of the design and value in pursuing a more rigorous evaluation. The validation for this project borrows from a rubric defined by Littenberg (1992), which describes five levels of technology assessment in medicine: plausibility, feasibility, intermediate outcomes, patient outcomes, societal outcomes [27]. Using this rubric, three aspects of the framework were validated: 1) plausibility (Does the framework appear valid?); 2) technical feasibility (Can the framework be implemented as a physical database model?); and 3) feasibility of the framework as a model in terms of completeness and usefulness of the model (Can the framework be used as a tool?).

In a formal evaluation, the third aspect (Can the framework be used as a tool?) would address three questions: 1) Is there a place in the framework for all meta data about processes, elements, and anomalies? 2) Is there more than one place where a single meta data element could be recorded? and 3) Are there places in the framework where no meta data elements would be recorded? These questions would be addressed in a formal evaluation looking at completeness, ambiguity, and non-useful features of the framework. In this validation, the goal is to show the practicality of the framework such as whether someone who has not worked with the framework can use it effectively and perhaps whether the terms make sense and the categories are used correctly, etc. The validation ends at the feasibility level in Littenberg's rubric although two types of feasibility were assessed, the technical feasibility (Can the framework be implemented as a physical database model?) and the feasibility of the framework by itself (Can the framework be used as a tool?). Future work could determine if the use and implementation of the framework would lead to intermediate outcomes perhaps in terms of less misinterpretation of data in health care databases. Using the framework to identify anomalies may also result in changes to processes and elements resulting in perhaps fewer numbers of errors, missing elements, etc. Patient and societal outcomes may very well result from improving the quality of care and reducing costs through the access and use of more reliable data in healthcare databases.

METHODOLOGY: SUBJECTS

There was no random selection or formal solicitation of subjects for this project. However, informally, two subjects were identified and asked to assist in the third aspect of the validation to help determine if the framework was practical to use. The two subjects were both involved with all aspects of acquiring, transforming, and analyzing data for a database used to capture data from a wide spectrum of GI Specialists throughout the United States. The database is called the Clinical Outcomes Research Initiative or the CORI dataset. It was important to find subjects who were already familiar with all aspects of a database due to the time constraints of having to learn about all the elements and the processes that generated those elements. Two subjects were asked to participate so that a comparison could be made between the differences in how the two used the framework.

METHODOLOGY: PROCESS

The framework was developed using training sets. These training sets consisted of the

UHC data analysis reports, the outcomes claims dataset, and a clinically rich database containing information on patients involved in emergency prehospital care incidents. In order to validate the framework, a testing set never before used in the development of the framework was sought. The importance of doing this was to help alleviate any bias that might result from using a dataset that was used in development, or was already familiar to the framework developer. As mentioned in the last section, the testing dataset was the CORI dataset.

The instruction given to the subjects was to: 1) select five to ten elements of their own choosing in the CORI database with a request for elements resulting from different processes; 2) record the elements and attribute meta data for the attributes they believed necessary for understanding the elements; 3) record the process steps responsible for generation of the selected elements and attribute meta data for the attributes they believed necessary for understanding the process steps; 4) record anomalies found in the actual CORI data; 5) record anomalies or potential anomalies the processes produce or may produce in the data; and 6) identify associations between the process anomalies and the element anomalies.

The subjects were given forms on which to record the meta data that were pre-numbered (see Appendix A for a sample set of the forms). The subjects were also given a half-hour training session which included a walk-through of two examples: the type of meta data needed on each form, and a completed set of forms (see Appendix A). In addition, the subjects received visual representations of the framework (see Figures 2 and 3), and two checklists one showing all element attributes and anomalies, and one showing process attributes and anomalies (see Appendix B). The attribute checklists were pre-numbered to ease filling out the forms. Finally, they were given a brief write up of the evaluation exercise which included a description of some of the less obvious process attributes and of all the anomalies types (see Appendix C). The subjects were given one week to complete the task.

RESULTS

Three aspects of the framework were validated: 1) Plausibility, Does the framework appear valid? 2) Technical feasibility, Can the framework be implemented as a physical database model? And, 3) the feasibility of the framework as a model, Can the framework be used as a tool?

DOES THE FRAMEWORK APPEAR VALID?

The result of a validation of plausibility is a subjective statement asserting that something is seemingly true. In this project, the assertion is that the framework seems valid by comparing the framework to the past and current understanding of the anomalies in health care databases from both research and the investigation of sample databases.

CAN THE FRAMEWORK BE IMPLEMENTED AS A PHYSICAL DATABASE MODEL?

Technical feasibility is demonstrated as a conceptual and physical design. The conceptual schema is shown in Figure 5. Figure 5 is a detailed diagram showing the relationships and entities necessary for formally implementing the framework as a series of tables in a relational database repository. In this diagram, the rectangles show the entities (processes, elements or anomalies) and the diamonds show how entities are related together. In a physical implementation, both rectangles and diamonds will translate into tables, but the use of these conventions helps to visualize the different functions the tables have. This diagram can then be translated into a series of tables, which makes up the physical schema. The physical schema with table descriptions and sample data is shown in Appendix D. The final physical schema was not implemented, but the initial physical schema based on the initial framework was implemented using an Oracle database. Data gathered from the assessment of the training databases were entered into the Oracle database and a web-based interfaced was developed to access those data. This proved to be useful in visualizing the framework and for discussions with others in discovering how the conceptual framework could manage real data. These discussions then resulted in refining the framework to better account for processes and their affects on elements and influence in the production of anomalies. From the new refined framework, another physical schema was then developed see Figure 5. In Figure 6, the schema was changed so that process attributes could be identified with the elements the attribute involved. For example, the event of entering a patient into the trauma system in a pre-hospital care environment involves designating a patient as requiring entry into the trauma system based on some set of criteria, tagging the patient with a trauma system identification arm band, etc. The attributes of the process for recording this event involves certain data elements including recording the criteria used to determine whether the patient required entry into the trauma system, the time they were entered, the trauma band identification number, etc.

In summary, the implementation of the initial framework was an Oracle database with a web access front end. The implementation of the refined framework is two physical design schemas (Figures 5 and 6), and a detailed layout of the tables, with examples, that would be created from these schemas (Appendix D).

CAN THE FRAMEWORK BE USED AS A TOOL?

The feasibility of the framework as a model was originally to be assessed by comparing the meta data received from two people who worked with the CORI dataset. The two sets of meta data were then going to be compared to remark on differences in how the framework was used. However, the two subjects ended up working on the analysis together producing one completed set of meta data forms rather than two (see a sample of the meta data forms in Appendix A). The requirement to have the two work separately was stated on more than one occasion, but was not written out. This along with the fact that the training occurred the week prior to the work being completed may have contributed to the misunderstanding. However, a significant sample of meta was received, and still provides valuable insight into the framework and considerations for future research. In addition, a team approach to gathering meta data is more representative of how analysts would perform this in a real work setting.

The completed set of forms that were received resulted in meta data for six elements in the CORI dataset, and their associated processes, and anomalies.

Results: Elements and Element Attributes Meta Data

The two subjects assessed six elements (see Table 1: Element Attributes). The forms included spaces for the following element attribute and the subjects were asked to record meta data for each of these attributes: physical name; datatype; length; and definition. All these attributes contained values. Of the remaining attributes, the subjects decided for themselves on which ones to record meta data. Five records included whether or not a default value existed. Four records included whether the element was required or not. Range values were entered on the three records where a range would have been appropriate and twice where the subjects used range to mean something other than a numeric minimum and maximum which was the original intent of that attribute. Instead, in the two records, the subjects used range to indicate that the value could only be true or false. Two records contained dependency attributes. Three records contained values for a structural attribute other than the ones listed on the checklist. The three uses of the "other" attribute where actually descriptions of process. Two attributes were descriptions of software features. The third attribute was really a description of two physical elements that together made up a conceptual element for denoting that a code for "indication" did not exist on the given list and so an "other indication" option has been recorded. The physical implementation in the CORI dataset consisted of a checkbox indicating that the "other indication" option was being used and a text field for recording a description of this "other indication."

TABLE 1. TABULATED RESULTS FOR ELEMENT ATTRIBUTES

Total Number of Elements Assessed: 6

ATTRIBUTE	TIMES USED
ELEMENT ATTRIBUTES	
Structural Attribute	
1) Physical Name	6
2) Datatype	6
3) Length	6
4) Precision	0
5) Format	0
6) Range (Min/Max Limits)	5
7) Required or Optional	4
8) Single code (mutually exclusive) or Multiple codes	0
9) Default Value	5
10) Other Structural Attribute	3
Definitional Attribute	
11) Definition	6
12) Other Definitional Attribute	0
Conditions and Consistency	
13) Conditionally dependent on	2
14) Consistent with	0
15) Other Conditions and Consistency Attribute	0

Results: Element Anomaly Meta Data

Four elements were recorded to have anomalies of an invalid value type (see Table 2: Element Anomalies). Two elements were recorded to have anomalies of missing value type. One element was recorded to have an anomaly of inconsistency type. One element was recorded to have two different anomalies of misrepresentation type. One anomaly description was actually two different anomalies one indicating an inconsistency error caused by users altering social security numbers. This would make previous records with that social security number inconsistent with later records.

TABLE 2. TABULATED RESULTS FOR ELEMENT ANOMALIES

ATTRIBUTE	TIMES USED
ELEMENT ANOMALIES	
a) Invalid Values	4
b) Valid values which misrepresent	2
c) Inconsistencies between values	1
d) Missing values	2
e) Multiple Anomaly Types	0
f) Other Anomaly Type	0

Results: Processes and Process Step Attributes Meta Data

The subjects recorded six process steps indicating that an individual process step matched each data element (see Table 3: Process Step Attributes). Four process steps shared three process attributes in common, so the subjects recorded the attributes once and referred to them on the other forms. The other two process steps also shared five process attributes in common, and again the subjects recorded the attributes once and referred to them on the other form. Five of the six major categories of process attributes were used: 1) who was involved in the process step; 2) what the description of the step was; 3) when the process step took place; 5) why the process step took place; and 6) how the process step was carried out. The major category not used was 4) where the process step took place. Six of the nine sub-attributes for categorizing how the process was carried out were also used which included the use of four acquisition attribute types and two transformation attributes: 6.B.) Measurement/observation; 6.C.) Record keeping practices; 6.D.) Constraints/rules; 6.E.) Changes in methods over time; 6.I.) Scrubbing algorithms; and 6.J.) Inclusion/exclusion criteria. No "other" types of process attributes were used.

TABLE 3. TABULATED RESULTS FOR PROCESS STEP ATTRIBUTES

Total Number of Elements Assessed: 6

ATTRIBUTE	TIMES USED
PROCESS STEP ATTRIBUTES	
Structural Attribute	
1) Who – Description of those involved in the process step	6
2) What – Description of the process step	6
3) When – Time when the process takes place	6
4) Where – Location where the process takes place	0
5) Why – The purpose or function of the process step	6
6) How – Methods used to complete the process step	
Acquisition Methods	
6.A.) Business Practices	0
6.B.) Measurements/Observations	1
6.C.) Record Keeping Practices	3
6.D.) Constraints/Rules	1
6.E.) Changes in Methods Over Time	1
6.F.) Other Acquisition Attribute	0
Transformation Methods	
6.G.) Extraction Rules	0
6.H.) Mapping Rules	0
6.I.) Scrubbing Algorithms	1
6.J.) Inclusion/Exclusion Criteria	1
6.K.) Other Transformation Attribute	0
6.L.) Other Method Attrubute	0
7.) Other Process Attribute	0

Results: Process Step Attribute Anomalies Meta Data

Four process anomalies were recorded for four different process steps (see Table 4: Process Step Anomalies). Two of the process anomalies were written from the viewpoint of the anomalies found in the data rather than how the process contributes to the production of the anomalies. The other two anomalies were keen observations about the effect element and process attributes (in this instance the effect of the element structure and entry interface structure) can have on the resulting data. Although these two anomalies were reported separately they are really related to each other. These two anomalies recognized that supplying a pre-determined list of indications may over-report some indications, the related anomaly recognized that if an "other" option was provided, indications on the checklist may be under-reported if "other" was used instead of an applicable item on the checklist. These last two anomalies were entered as having multiple anomaly types, but should have been entered as a misrepresentation anomaly type.

TABLE 4. TABULATED RESULTS FOR PROCESS STEP ANOMALIES

PROCESS STEP ANOMALIES	
a) Invalid Values	0
b) Valid values which misrepresent	0
c) Inconsistencies between values	0
d) Missing values	0
e) Multiple Anomaly Types	4
f) Other Anomaly Type	0

Results: Association of Process Attribute Anomalies to Element Anomalies

The results recorded on the form to associate anomalies did not make sense, and it is assumed that the subjects were either too hurried or confused about the form to enter the items correctly. No attempt was made to go back to the subjects to answer why the form entries were invalid. Even though the reason is unknown as to why accurate meta data were not recorded on the form, it provides a trigger for questioning either the design of the forms, the delivery of the training, or the framework itself. The importance of these results will be discussed in the following section.

DISCUSSION

This discussion addresses the three aspects of the framework that were validated: 1) the plausibility of the framework; 2) the technical feasibility of the framework; and 3) the feasibility of the framework as a model in terms of completeness and usefulness of the model.

DOES THE FRAMEWORK APPEAR VALID?

Plausibility shows that the framework has some grounding in research and current applications, and was not invented as something clearly irrelevant and unsupported. The comparison of the framework to past research and current datasets was found to be very useful in supporting the assertion of plausibility. It shows that the anomaly data, process data and element data found in research and the sample databases can be modeled using the framework. This means that if known anomaly data, process data and element data can be modeled then it is very likely that any future discoveries of anomalies and characterization of process and element data can be modeled in the framework as well. Limitations exist in that the number of datasets assessed was three and the research in this area is relatively scant, so the amount data to compare the framework to was in no way exhaustive. However, the framework provides a basic model of the relationship of processes and elements and the production of anomalies and can be used in future assessments that will serve to refine the framework further.

CAN THE FRAMEWORK BE IMPLEMENTED AS A PHYSICAL DATABASE MODEL?

A physical schema describing table structures and table relationships to support the framework shows how real meta data may be stored physically and subsequently accessed. Although only the initial framework and subsequent physical design was actually implemented, it was found to show where the application of actual data to the framework either fit or didn't fit leading to subsequent refinement of the framework and the physical schema.

In addition, the implementation made the conceptual model more concrete to the analysts giving feedback on the framework. Having a physical schema that can be implemented means that a practical implementation of the framework is possible, and that it was useful in understanding not only the data, but the framework as well.

There were some limitations to developing a detail physical schema. Translating a high level model such as the one shown in Figure 3 into a physical schema wasn't straightforward, and attempts to make it flexible enough to accommodate all relationships between processes and elements, elements and elements and even processes and processes resulted in increasing complexity. There is most likely some point at which the technical complexity of attempting to allow for all types of meta data exceeds any usefulness a physical implementation in the form of a repository may offer in that the effort to capture these meta data may be more than users wish to expend. On the other hand simplifying the schema may result in losing the ability to model relationships accurately.

Finally, those using the framework may construct meta data in ways a physical schema may not support. In fact, this is what happened when the subjects were validating the framework. The subjects identified the conceptual data element of "other indication" as consisting of two physical elements; one being a check box indicating whether an "other indication" was selected, and the other being a free text field in which to describe the indication. Redesign of the physical schema to show conceptual elements as consisting of one or more physical elements was considered in one of the earlier versions. However, this seemed to complicate the design and was not included, yet this is in fact how these subjects organized their data. However, it is not known whether others would think the same way.

A second design consideration came when the subjects attempted to map each process step to an individual element. The original intent of the framework was to identify process steps and then the process attributes would address constraints or other affects on individual elements. The goal was to map anomalies the processes produced to anomalies elements suffered from. The schema as shown in Figure 5 would not have explicitly shown a process attribute's relationship to the individual elements it involved (see Figure 6 for a possible update to the schema where process attributes can be identified with the elements the attribute involves). However, characterizing meta data in this way may be important, at least to some people, in better understanding these data, processes, and their relationships.

Design considerations such as these may very well be design features that would be included or not depending on the particular implementation, and the schema shown in Figure 5 would serve more or less a template. The physical schema layout (Appendix D) then becomes more of an example of possible field definitions and sample data for constructing an implementation.

CAN THE FRAMEWORK BE USED AS A TOOL?

This validation exercise is a proof of concept that addresses the usefulness of the framework. Probably the most encouraging result found was the ability of the subjects to use a variety of the attribute types according to the proposed definitions, and few of the attribute types labeled "other." Another encouraging result is that for only six 67% of the element attribute categories were used, and 71% of the process attribute categories were used (both these calculations excluded the "other" type categories). This could indicate that the lists of attribute categories for both process attributes and element attributes are clear and fairly complete. This would give some support for answering the question, "Is there a place in the framework for all database meta data?" The fact that the subjects were able to use several categories in the framework effectively for only a small number of variables may also support the validity of the framework as a useful tool.

There was some concern that data was recorded in more than one place. In the element meta data, the subjects selected the "other" option for entering data which were by definition process meta data in terms of how the software either constrained the element or how it used the element to trigger the creation of other elements. This may indicate some misunderstanding of what a process is in that a process includes both human and machine (including software) attributes. In addition, the subjects identified a single process step with each data element, which may in fact be what most people would do and may reduce the ability to consider how processes affect data on a less concrete and a grander scale. For example, in analyzing the prehospital dataset, it was discovered the process of allowing chart reviewers to know whose charts they were reviewing and the chart authors to know who was reviewing their charts, along with the possibility of rank differences, lead to chart reviews of lesser quality possibly increasing the anomalies in the data. In this example, there was no direct relationship of process step attribute to an element; however, the process was so important to consider and it was actually found to adversely affect the data that the organization changed the process of chart review. Rather than a failure of the framework, though, there may be a need for better education in what process is and the relationship of process to the data elements.

Limitations with this validation are mostly concerned with the broad nature of the categories, in that it is easier to fit data into broad categories. Broad categories may be useful for triggering thought on what possible attributes may result in producing or contributing to anomalies. However, it may be more useful to try to catalog specific anomalies and the process attributes that produce them and element attributes that influence them.

By cataloging specific anomalies and how they are produced, it may be discovered that some types of process attributes and element attributes actually contribute to predictable types of anomalies. For example, the subjects discovered that by providing a list of indications, an over-reporting of some indications occurred. This could indicate that pre-determined lists of any kind might result in the over-representation of certain selections. On the other hand, providing an "other indication" option may result in an under-reporting of an indication on the list that should have been selected but wasn't such as when users prefer to enter their own description rather than take to time to find it on a list.

In the prehospital care dataset, users could enter a full or partial medication name and the system would attempt to find it in the medication database. However, if the medication was not found, such as if it was spelled incorrectly, the system would allow the user to accept the entry. Unfortunately, the effort to make the program more usable for users sometimes resulted in the same medication existing in the database by many different spellings.

These attributes of elements and processes may in fact provide some prediction for what anomalies may be produced and to what extent. If a data element is optional, in that the user is not required to enter information either by program or procedural constraint, then the chance for missing data may be greater than if the user was required to enter data. Although this is not necessarily the case, there may be a predisposition to this happening. On the other hand requiring a data element, especially by programmatically forcing entry, may result in more data being invalid or misleading as users attempt to satisfy program requirements rather than ensure data correctness.

The structure of the elements and process attributes may very well create tendencies toward certain anomalies. However, changing either the structure or process may not correct or prevent anomalies. In this instance, understanding not only the process, but also the characteristics of the business and people involved in acquiring the data becomes very important in understanding why anomalies occur. For example, users may select not to enter data into an optional field because it is not relevant, they don't have time, they don't want to for some reason, etc. A catalog of specific attributes and the possible anomalies they could produce would offer those using the framework more concrete examples and could be a valuable tool not only for analysts, but for software developers, and those managing and involved in data acquisition and transformation.

Certainly this validation is not a replacement for a formal evaluation although a formal evaluation may be difficult. If this method for validation were used on a larger sample of subjects, it would be difficult to control for variables such as how much knowledge the subjects had about a particular dataset and the processes used to created it, or differences in types of databases, etc. Certainly mock datasets and narratives of the processes could be used to control for some of the variables. However, it may be more useful to use the framework to discover specific types of anomalies attempting to construct a more complete description of how processes affect data.

APPLICATION OF THE MODEL

A real world application of the model would be an investment in time and money in not just setting up a meta data repository, but in the ongoing work that would need to be done to keep it updated especially as processes and even element attributes change and as new discoveries are made. However, there are potential benefits of utilizing this framework in this way. By providing this meta data about attributes and anomalies, the time formerly taken by analysts to uncover this information would be saved. In addition, as this type of meta data becomes available to a wide range of people, it could result in those lacking the resources to compile the meta data to actually use these data more than if they did not have this meta data. As anomalies are discovered and the reasons for their existence are understood, fewer anomalies may develop or the number reduced from such measures as changing the processes or even just by educating those taking the measurements and recording the data on how to reduce the occurrence of anomalies. Finally, this type of meta data could benefit designers on how to better design systems and develop processes in order to reduce the production of anomalies.

CONCLUSIONS

The purpose of this work was to develop a framework that can be used to characterize data and identify anomalies in health care databases. In essence, the framework is a model for thinking about data and how it is related to the processes that create it. These processes and the attributes of these processes really are the pivotal point for understanding these data. Who generates these data; why these data are generated; where they are generated; when they are generated; how data are generated and how data are affected by these processes ultimately influence the errors, omissions, inconsistencies, and misrepresentations which occur in these data.

There is certainly more discovery that needs to be made in the relationships of process and elements. Yet simply raising the awareness of these relationships and their effect on data is an important step towards improving health care data.

REFERENCES

1. Krowchuk, H.v., M.L. Moore, and L. Richardson, *Using health care records as sources of data for research.* Journal of Nursing Measurement, 1995. **3**(1): p. 3-12.

2. Paul, J.E., K.A. Weis, and R.A. Epstein, *Data bases for variations research.* Medical Care, 1993. **31**(5 Suppl): p. YS96-102.

3. Tierney, W.M., *et al.*, *Predicting inpatient costs with admitting clinical data.* Medical Care, 1995. **33**(1): p. 1-14.

4. Scully, K.W., *et al.*, *Development of an enterprise-wide clinical data repository: merging multiple legacy databases.* Proc AMIA Annu Fall Symp, 1997: p. 32-6.

5. Guralnik, D.B., ed. *Webster's New World Dictionary of the American Language.* Second College Edition ed. . 1979, William Collins Publishers, Inc.: Cleveland.

6. Connell, F.A., P. Diehr, and L.G. Hart, *The use of large data bases in health care studies.* Annual Review of Public Health, 1987. **8**: p. 51-74.

7. Dolin, R.H., *Outcome analysis: considerations for an electronic health record.* MD Computing, 1997. **14**(1): p. 50-6.

8. Lee, J.Y., *Uses of clinical databases.* American Journal of the Medical Sciences, 1994. **308**(1): p. 58-62.

9. Lee, T.H. and L. Goldman, *Development and analysis of observational data bases.* Journal of the American College of Cardiology, 1989. **14**(3 Suppl A): p. 44A-47A.

10. Edwards, F.H., R.E. Clark, and M. Schwartz, *Practical considerations in the management of large multiinstitutional databases.* Ann Thorac Surg, 1994. **58**(6): p. 1841--1844.

11. Sorensen, H.T., S. S., and O. J., *A framework for evaluation of secondary data sources for epidemiology research.* International Journal of Epidemiology, 1996. **25**(2): p. 435-442.

12. Cimino, J.J. *Desiderata for controlled medical vocabularies in the twenty-first century.* 1997. New York: Department of Medical Informatics, Columbia University.

13. Das, A.K. and M.A. Musen, *A foundational model of time for heterogeneous clinical databases.* Proc AMIA Annu Fall Symp, 1997: p. 106-10.

14. Johnson, S.B., T. Paul, and A. Khenina, *Generic database design for patient management information.* Proc AMIA Annu Fall Symp, 1997: p. 22-6.

15. *Discharge Abstract Verification Report*, May, 26. 1998, University HealthSystem Consortium: Oak Brook,

16. Chowdhury, S., *et al., Methods for knowledge extraction from a clinical database on liver diseases.* Computers & Biomedical Research, 1991. **24**(6): p. 530-48.

17. Arndt, S., *et al., Effects of errors in a multicenter medical study: preventing misinterpreted data.* J Psychiatr Res, 1994. **28**(5): p. 447-459.

18. Prather, J.C., *et al., Medical data mining: knowledge discovery in a clinical data warehouse.* Proc AMIA Annu Fall Symp, 1997: p. 101-5.

19. Sackett, D.L., *Bias in Analytic Research.* Journal of Chronic Diseases, 1979. **32**: p. 51-63.

20. Sasse, M.A. and D. Fulton, *Systems Analysis and Design*, . 1998, Department of Computer Science -- University College London: London. p. 1-19. Online Report -- http://www.cs.ucl.ac.uk/staff/A.Sasse/b123/b123book/chp8/chp8-.html.

21. Wyatt, J.C., *Clinical data systems, Part 1: Data and medical records [see comments].* Lancet, 1994. **344**(8936): p. 1543-7.

22. Tsai, Y.S., *et al., An expert-guided decision tree construction strategy: an application in knowledge discovery with medical databases.* Proc AMIA Annu Fall Symp, 1997: p. 208-12.

23. Wagner, M.M. and W.R. Hogan, *The accuracy of medication data in an outpatient electronic medical record.* Journal of the American Medical Informatics Association, 1996. **3**(3): p. 234-44.

24. Hersh, W.R., *The electronic medical record: Promises and problems.* Journal of the American Society for Information Science, 1995. **46**(10): p. 772-776.

25. Harris, H.E., *et al.*, *Methodological considerations in the design of an obstetric database abstracted from medical records.* Methods of Information in Medicine, 1997. **36**(3): p. 191-200.

26. Brackett, M.H., *The data warehouse challenge: Taming data chaos.* 1996, New York: John Wiley & Sons, Inc.

27. Littenberg, B., *Technology assessment in medicine.* Academic Medicine, 1992. **67**(7): p. 424-428.

FIGURE CAPTION

FIGURE 1. Process Model

FIGURE 2. Initial Framework

FIGURE 3. Framework Model

FIGURE 4. Framework Model with Attributes

FIGURE 5. Physical Schema

FIGURE 6. Alternate Physical Schema

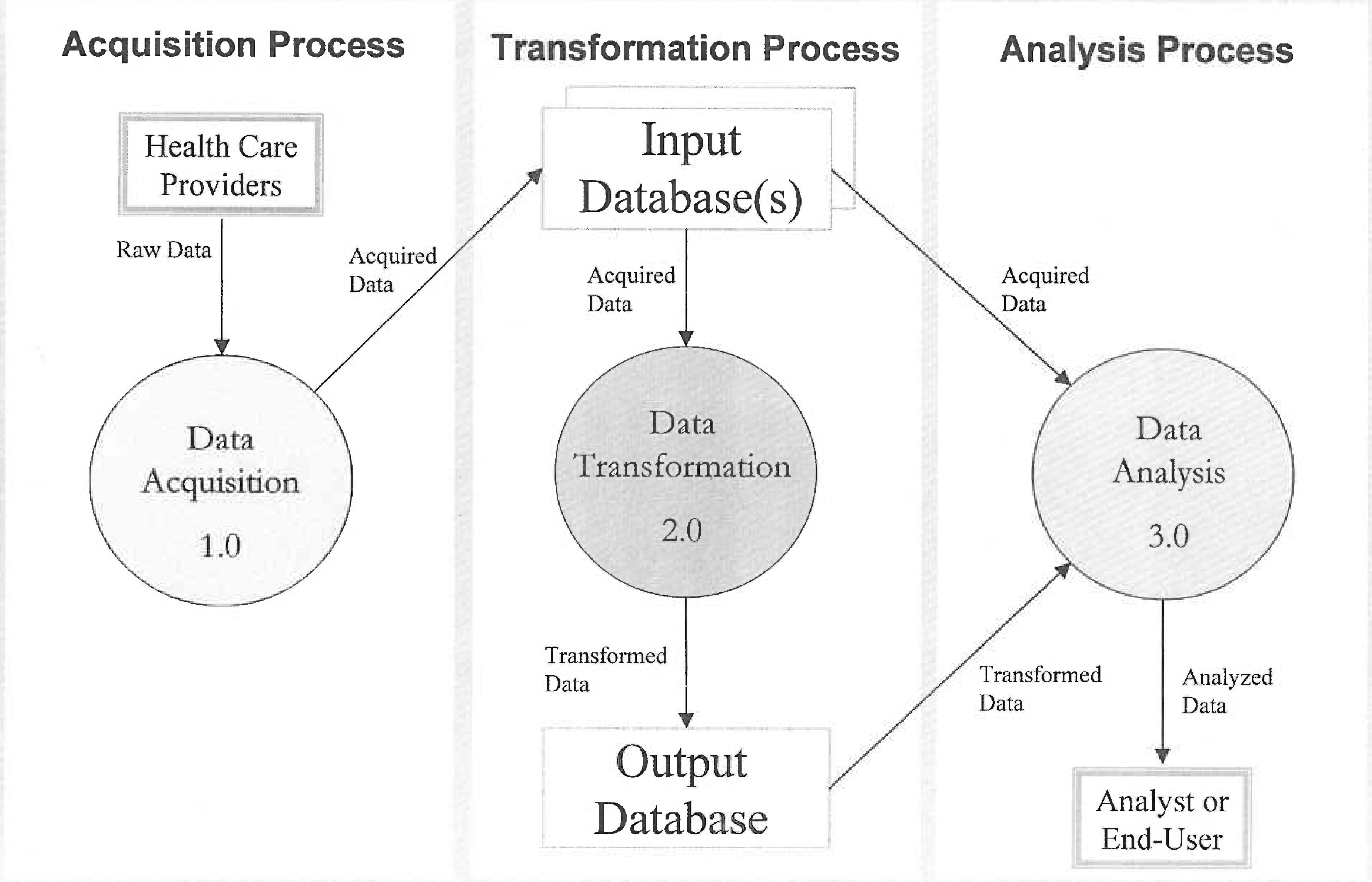

Figure 1. Process Model

	PROCESS		
	Acquisition	**Transformation** (zero to many)	**Analysis** (zero to many)
	Potential Causes of Anomalies		
Errors	Data Entry Invalid Value Misrepresentation Consistency Other	Transformation Algorithms Other	Wrong Assumptions Other
Missing	Not Applicable Unknown Not Measured Not Noteworthy Not Recorded Mis-Recorded Lost Data Assumed Normal Values Other	Exclusion of Rows/Records Exclusion of Fields Exclusion of Values Other	Eliminating rows/Records Eliminating Fields Eliminating Values Other
Variation	Structural Definition Method / Measurement Constraints/Rules Temporal/Historical Record Keeping Practices Business Practices Purpose Other	How Data are Transformed Other	How Data are Used Assumptions Other

Figure 2. Initial Framework.

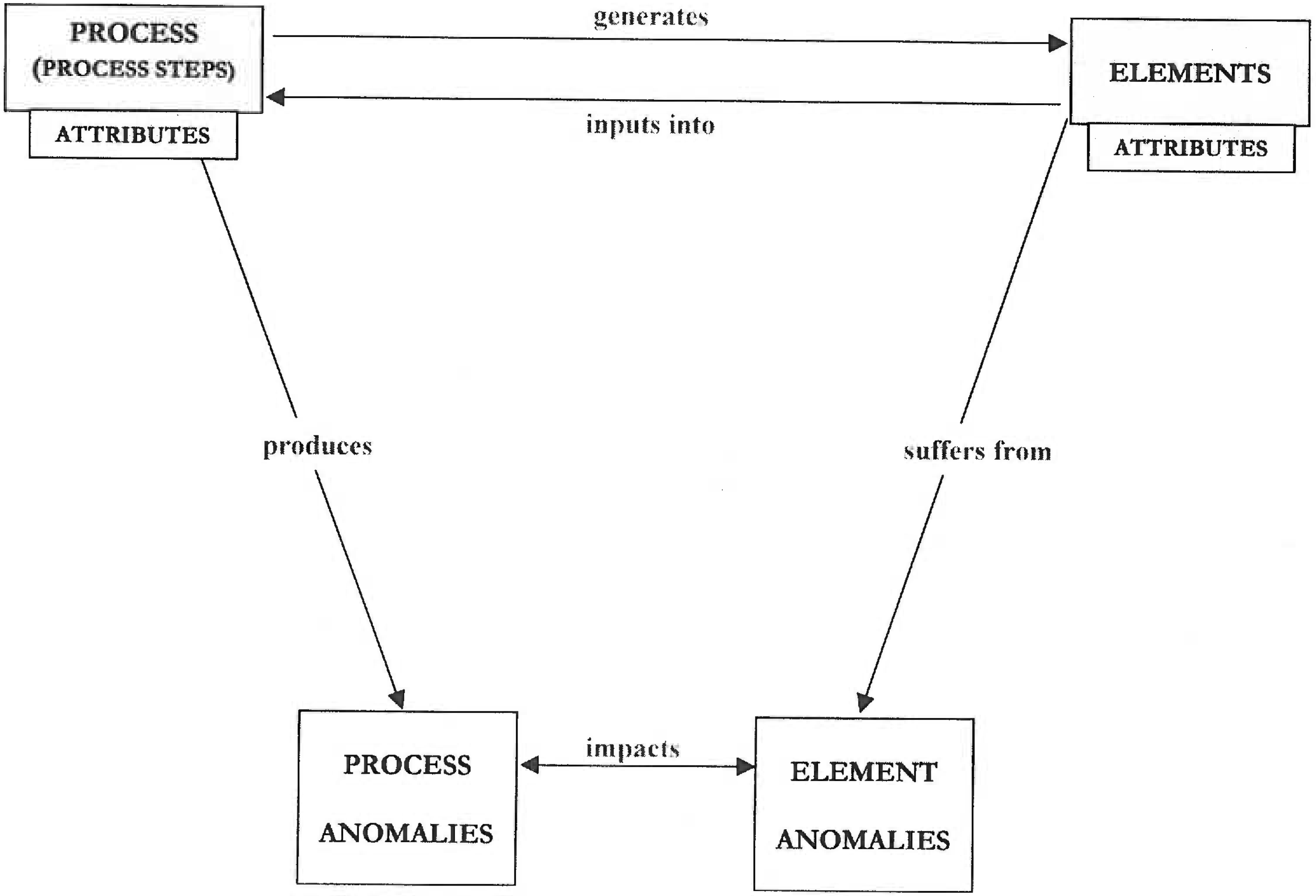

Figure 3. Framework Model

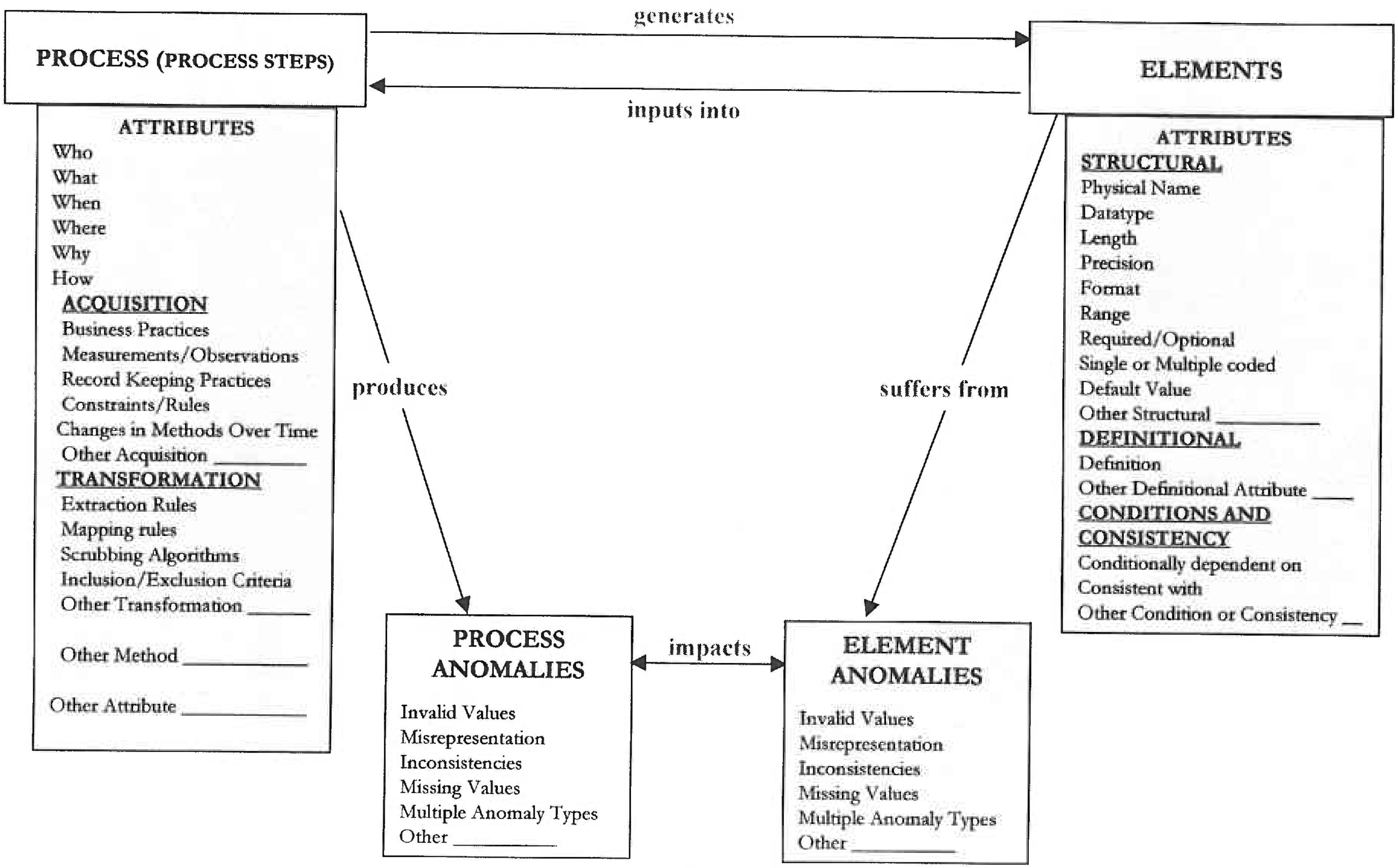

Figure 4. Framework Model with Attributes

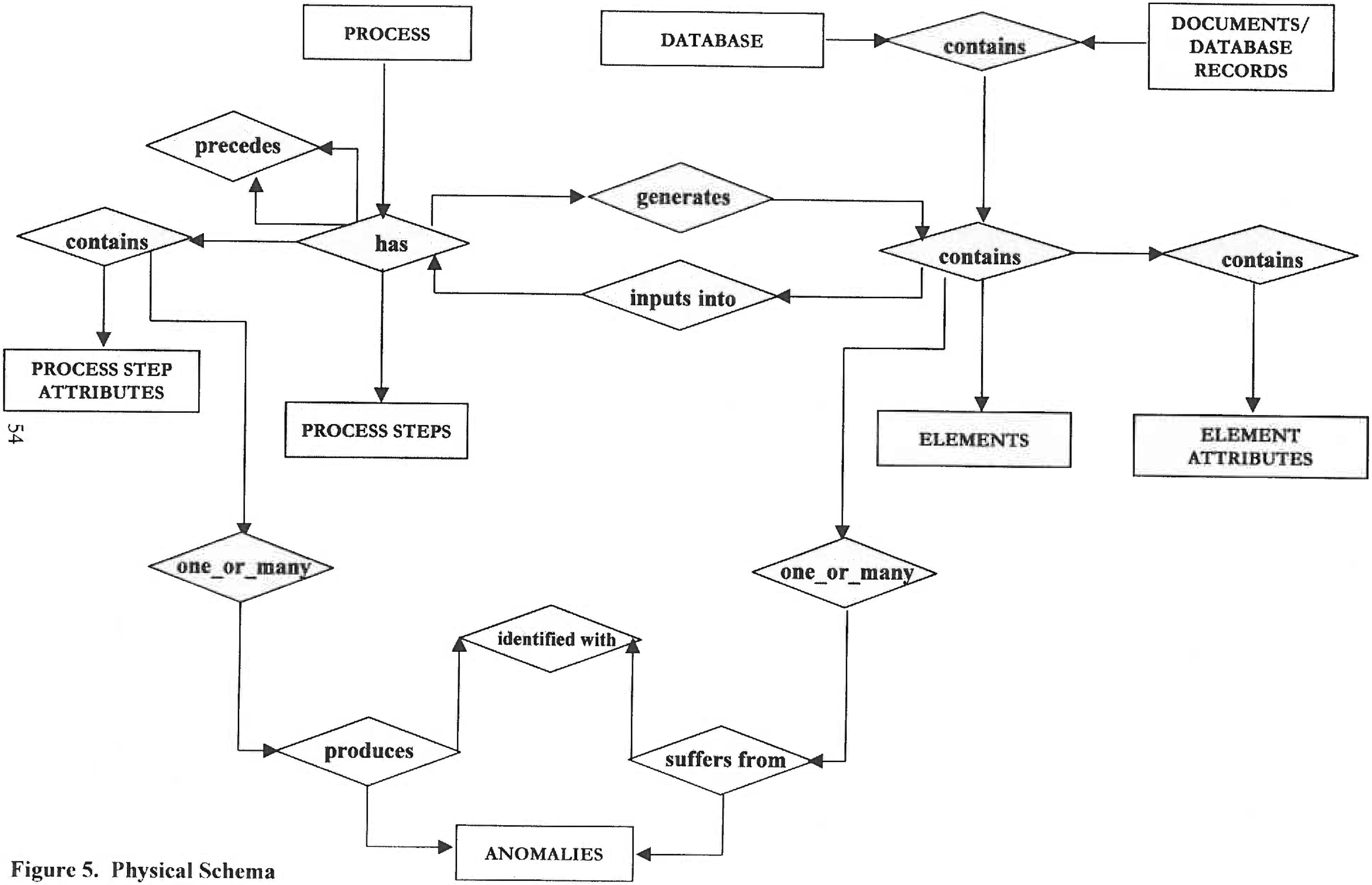

Figure 5. Physical Schema

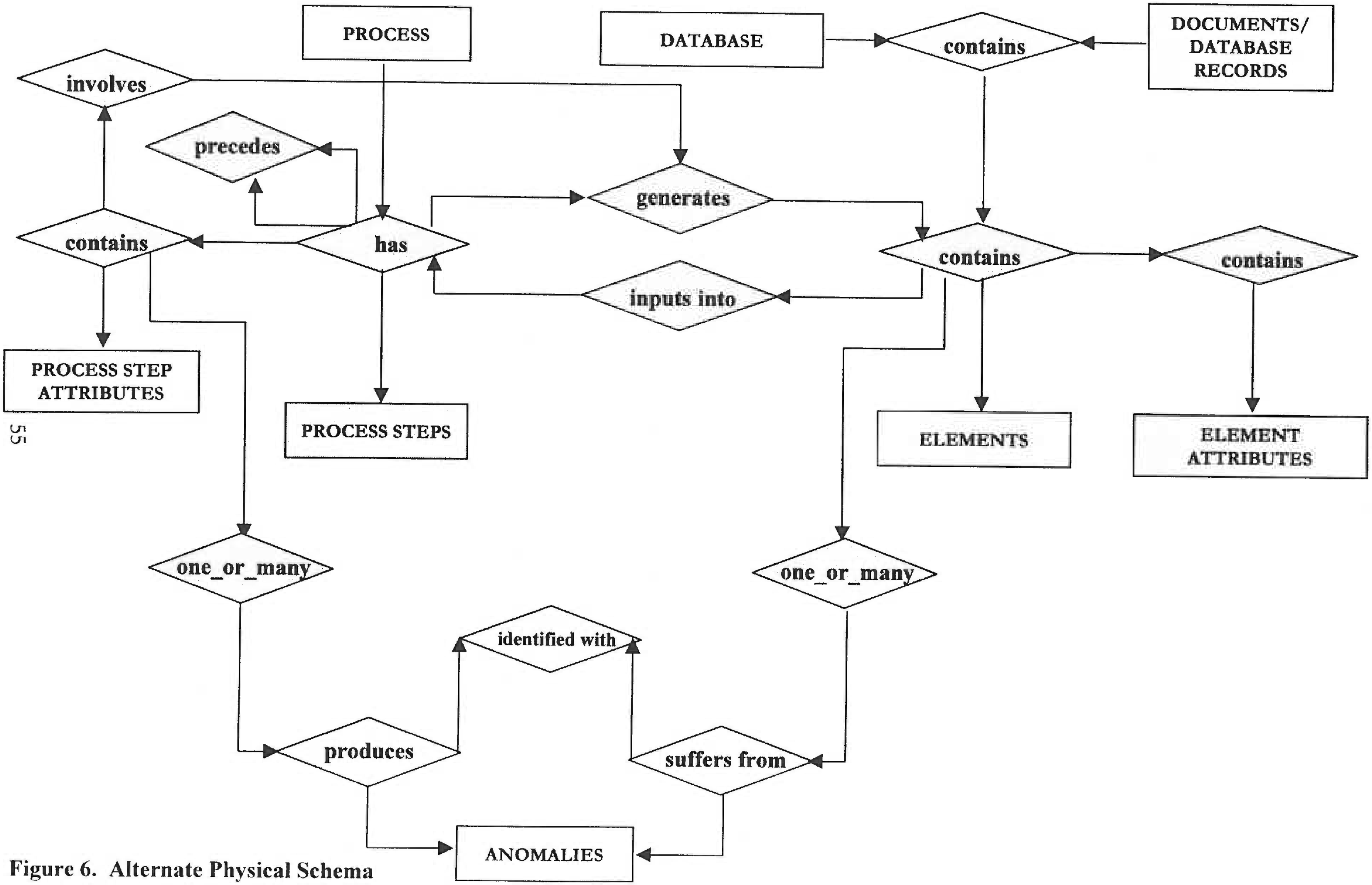

Figure 6. Alternate Physical Schema

APPENDIX A

VALIDATION FORMS WITH SAMPLE DATA

Forms have been slightly truncated to fit into thesis format.

I. _*PREHOSPITAL CARE REPORT STATUS CODE*___________ ELEMENT ENTITY

NO.	Attribute NO. (Desc. if "other")	Value
Ex.	(1) physical name	patient_disposition_code
1	(1) physical name	*status_code*
2	(2) Datatype	*varchar*
3	(3) Length	*2*
4	(11) Definition	*Shows the status of the prehospital care patient report. Codes: 'XX' = No Status; 'IN' = Report Initiated; 'PP' = Pending Partner Review; 'PS' = Pending Station Review.*
5	*(7)*	*Required*
6	*(8)*	*Single Code*
7	*(9)*	*'XX'*
8	*(13)*	*Status 'PP' depends on all required fields having values. Status 'PS' and 'CO' depend on a reviewer paging through all report screens and answering "yes" to 'Report Complete' prompt.*
9		
10		

II. ___ *PREHOSPITAL CARE REPORT REVIEW* ______ PROCESS STEP ENTITY

NO.	Attribute NO. (Desc. if "other")	Value
Ex.	6.D	*New codes are added and old codes are deleted from code tables as needed. There are no dates to identify when a new code has been added or when an old code has been deleted or is no longer used.*
1	*(1)*	*The partner first reviews the report then an in-station reviewer who is a certified paramedic of any rank who has been appointed by the station caption and is known to all station personnel.*
2	*(2)*	*Review of prehospital report to check for errors and incompleteness.*
3	*(3)*	*Occurs after the author completes prehospital report and all required fields have values.*
4	*(4)*	*Review can be performed from any terminal; however, the reviewer must be logged into his or her own account.*
5	*(5)*	*To provide a quality check for reports and reduce errors, missing data and misrepresentation of the incident.*
6	*(6.A)*	*Once the report is completed the author's designated partner reviews the report. The partner reviews by paging through the all the screens after which they are asked if the report review is complete, if 'yes' then the report goes to the station reviewers pending review list.*
7	*(6.A)*	*In-station review is performed by a station reviewer(s) who reviews all EMS reports for the entire station. The review process is the same as for the partner reviewer. The in-station reviewer can send the report back to the author for revision. Only the author can make revisions to the report.*
8	*(6.D)*	*Although the program cannot force a reviewer to read the report, it does force the reviewer to page through every screen in the report at least once before the reviewer is able to sign off.*

III. _*PREHOSPITAL CARE REPORT NARRATIVE*____________ ELEMENT ENTITY

NO.	Attribute NO. (Desc. if "other")	Value
Ex.	(1) physical name	patient_disposition_code
1	(1) physical name	*blob_text*
2	(2) Datatype	*text*
3	(3) Length	*N/A*
4	(11) Definition	*Free-text narrative for prehospital care incidents. Contains limited SOAP information. Mainly subjective.*
5	*(7)*	*Required*
6		
7		
8		
9		
10		

IV. _*PREHOSPITAL CARE REPORT INTERVENTIONS*___________ ELEMENT ENTITY

NO.	Attribute NO. (Desc. if "other")	Value
Ex.	(1) physical name	patient_disposition_code
1	(1) physical name	*item_value, item_code, uom_code*
2	(2) Datatype	*varchar*
3	(3) Length	*various*
4	(11) Definition	*A generic field containing all intervention values. The user defines all interventions, also intervention codes include assessment and other procedure data.*
5	*(7)*	*Required to have only the PIC identified and a basic ABC assessment. Entry of other interventions depend on the incident and is at the discretion of the paramedic; however, every procedure and intervention performed at the scene should also be recorded in the record.*
6	*(8)*	*Depends on type of intervention.*
7	*(9)*	*Depends of type of intervention*
8	*(13)*	*Depending on the intervention code used, some interventions are dependent on others.*
9	*(14)*	*The time element associated with an intervention must not precede the arrival date to the scene. If an intervention was performed prior to arrival by the ambulance the letters 'PTA' are entered into the time field.*
10		

PROCESS ANOMALIES

P-NO.	NO.	Attribute NO. (Desc. if "other")	Anomaly Type	VALUE
IV.	Ex.	6.D	(a)	*Because it is unknown when new codes are added and old codes are deleted or no longer being used, codes added or deleted within a time from being studied may result in some codes being under-represented for the entire time frame.*
(II)	1	***(1)***	***(e)***	***Since a reviewer may rank below the author, and since peer pressure to give a good review to fellow employees combined with the knowledge to all personnel who the reviewer is and to the reviewer who the author is, there is evidence that report reviews are not stringent enough resulting in reports which are less accurate and less complete than they should be.***
	2			
	3			
	4			
	5			

ELEMENT ANOMALIES

E-NO.	NO.	Anomaly Type	VALUE
III.	Ex.	(a)	*There is a drastic drop in trauma system entries based on patient disposition for 1997. It appears that the patient disposition code for "Trauma System Entry" was discontinued sometime in early 1997.*
(I)	1	***(d)***	***Although reports indicate they are complete, a review of data elements suggests many elements may contain no value where the type of incident indicates a value could have been recorded.***
(III) ***(IV)***	2	***(c)***	***A comparison of narratives to intervention records within a patient report indicate that there are inconsistencies between the two.***
	3		
	4		
	5		

ANOMALY ASSOCIATION

PROCEDURE ANOMALY NO.	ASSOCIATION	ELEMENT ANOMALY NO.
Example procedure anomaly	identified with	Example element anomaly
(1)	identified with	*(1)*
(1)	identified with	*(2)*
	identified with	
	identified with	
	identified with	
	identified with	
	identified with	
	identified with	
	identified with	
	identified with	

APPENDIX B

PROCESS AND ELEMENT CHECKLISTS.

CHECKLIST FOR PROCESS STEPS

ENTITY	ATTRIBUTE	VALUE
	(1) Who – People involved in the Process step	
	(2) What – Description of Process Step	
	(3) When – Time Process Step Takes Place	
	(4) Where – Location Process Step Takes Place	
	(5) Why – The Purpose or Function of the Process Step	
	(6) How – Methods Used to Complete the Process Step Acquisition (6.A) Business Practices (6.B) Measurements/Observations (6.C) Record Keeping Practices (6.D) Constraints/Rules (6.E) Changes in Methods Over Time (6.F) Other Acquisition ____________ Transformation (6.G) Extraction Rules (6.H) Mapping Rules (6.I) Scrubbing Algorithms (6.J) Inclusion/Exclusion Criteria (6.K) Other Transformation _________ (6.L) Other Method ________________	
	(7) Other Attribute __________________	
	ANOMALIES	
	(a) Invalid Values (b) Valid Values which Misrepresent (c) Inconsistencies between Values (d) Missing Values (e) Multiple Anomaly Types (f) Other __________________	

CHECKLIST FOR ELEMENTS

ENTITY	ATTRIBUTE	VALUE
	Structural	
	(1) Physical Name	
	(2) Datatype	
	(3) Length	
	(4) Precision	
	(5) Format	
	(6) Range(Min/Max Limits)	
	(7) Required or Optional	
	(8) Single code (mutually exclusive) or Multiple codes (ordinality)	
	(9) Default Value	
	(10) Other Structural Attribute ____________	
	Definitional	
	(11) Definition	
	(12) Other Definitional Attribute __________	
	Conditions and Consistency	
	(13) Conditionally dependent on	
	(14) Consistent with	
	(15) Other Conditions and Consistency _____	
	ANOMALIES	
	(a) Invalid Values	
	(b) Valid Values which Misrepresent	
	(c) Inconsistencies between Values	
	(d) Missing Values	
	(e) Multiple Anomaly Types	
	(f) Other ____________________	

APPENDIX C

EVALUATION INSTRUCTIONS

WEDNESDAY, JANUARY 20, 1999

FRAMEWORK EVALUATION TRAINING

AGENDA

BACKGROUND

THESIS

Framework for characterizing data and identifying anomalies in health care databases.

PURPOSE

To address issues concerned with the use of data in healthcare databases.

Issues regarding the quality, accuracy, and completeness of the data.

FRAMEWORK

To identify problems with quality, accuracy, and completeness within a database, a framework was developed to organize one's thinking about databases and the processes that create them.

SEE FRAMEWORK

Within the framework, categories identify key areas where data about data (meta data) may be important in developing an understanding about databases and the quality, accuracy, and completeness of the data within them.

SEE FRAMEWORK WITH CATEGORIES

EXAMPLES

CATEGORY EXAMPLES

Acquisition

BUSINESS PRACTICES Some business practices will affect what data are available for analysis. For example from the UHC analysis report: eight hospitals do not provide services to maternity cases; one hospital did not provide services to maternity cases prior to 1/1/96.

MEASUREMENT/OBSERVATIONS Method variation covers a wide area since the ways data are collected, recorded, edited and stored may vary greatly. These differences may include human versus machine collection of data, or differences in how measurements are made. For example, whether pulse is counted for an entire minute or some fraction then multiplied out. If one always counts pulse for 6 seconds then multiplies by 10, the resulting value will always be a factor of 10.

RECORD KEEPING PRACTICES Some record keeping practices may affect how the data can be used during analysis. Consider the following: Two hospitals (in the UHC analysis report) report room and care charges for some newborns on the mother's record assuming that when these babies are healthy, they will have very low charges per day. More from the UHC report: three hospitals do not provide ICD-9-CM codes for a number of diagnostic procedures that are performed at their facilities; one hospital does not provide ICD-9-CM codes for a number of non-billable procedures.

CONSTRAINTS/RULES Constraints variation can best be defined as those rules that are applied to data either through human enforcement or software enforcement in order to restrict or constrain what values may be entered into a field. This may include criteria used to decide what values to enter such as standard coding practices, to relationships which are enforced within the data, to default values which are set, etc. An example would be limits or ranges for certain data values. Another example would be the set of codes from which an individual may select. Default values may be of special concern since it may be quite easy for someone to accept a default value. In this case knowing what fields have defaults, what the default values are, and then knowing how often the default occurs may be very important to analysts.

CHANGES IN METHODS OVER TIME Temporal variation really refers to the changes that may occur in any of the above types of variation within a single database or system over time. The changes really cause a difference between the data collected before the change and the data collected after the change. For example, the UHC (University HealthSystem Consortium) database, with compiled data from several university hospitals, reported that OHSU's submitted database had used codes that no longer existed and were now invalid when compared with the new set of codes. Another example from UHC states that in order to locate the appropriate DRG name in the DRG name file, it is necessary to match by DRG and DRG version. This is because the same DRG number can have a different definition and description for different discharge dates.

Transformation

EXTRACTION RULES The rules used to extract the data from the source databases and files.

MAPPING RULES Any process used to reconcile differences between process attributes and element attributes.

SCRUBBING ALGORITHMS Any process for cleaning the data (ie. removing errors, converting formats).

INCLUSION/EXCLUSION CRITERIA Rules for including or excluding records from the resulting database or data set (ie. excluding records with missing values, or ones where a field value does not match the end database format such as a character field to numeric field).

ANOMALIES

INVALID VALUE

MISREPRESENTATION The value is correct, but does not reflect the true situation. For example, if only one code is allowed to describe a situation where multiple codes would better describe a situation, the situation may not be adequately described by the one code. Misrepresentation may become more pronounced when trying to merge databases. Attempting to create a common set of codes and assuming a common process and understanding of the data may easily result in a merged database that is not representative of the original data or patients.

INCONSISTENCY A data value may be valid and accurately represent the situation or observation; however, when considered with other values there are inconsistencies. For example, if the definitions of codes are unclear one person may use one code to describe a situation and another person may use a difference code to describe the same situation. This problem will not only create skewed results when trying to analyze the value, but may be difficult to discover. Inconsistencies may also occur when the value for one or many data elements is not appropriate based on values for other data elements. For example, if a patient condition is described as pregnant yet the patient demographics describe the patient as male then an inconsistency has probably occurred in one of the two elements.

MISSING VALUES

MULTIPLE ANOMALY TYPES

FRAMEWORK EXAMPLES

STANDARD FORM EXAMPLE

FILLED IN FORM EXAMPLE

DATA ANALYSIS

- Please complete the analysis using the forms provided. I have additional forms if you need them.
- The time period for the analysis is from Wednesday, January 20, 1999 at 3:00 to Friday, January 22, 1999 at 3:00.

IF YOU HAVE ANY QUESTIONS PLEASE CONTACT ME BY E-MAIL AT:

savagean@ohsu.edu

I WILL ALSO BE IN THE BUILDING ON THE 5TH FLOOR CUBICLE 526X FOR THE TIME PERIOD OF THE ANALYSIS

THANK YOU

APPENDIX D

PHYSICAL SCHEMA

PHYSICAL SCHEMA

DEFINITIONS AND EXAMPLES

PROCESS

DEFINITION

A process is a group of steps for acquiring, transforming, or analyzing data.

TABLE

PROCESS		
FIELD	**DESCRIPTION**	**EXAMPLE**
ProcessID	Unique Identifier	
Name	Name of the Process	Prehospital Care Emergency Response Reporting
Description	Description of the Process	
Process Type	The process' main function is for the **acquisition**, **transformation**, or **analysis** of data.	Acquisition
DateTimeCreated	Date and time row was created	
DateTimeUpdate	Date and time row was last updated.	
UpdatedBy	Who updated the row	

PROCESS_STEP

DEFINITION

Ordered step in the process of acquiring, transforming, or analyzing data.

TABLE

PROCESS_STEPS		
FIELD	**DESCRIPTION**	**EXAMPLE**
ProcessStepID	Unique Identifier	
Name	Name of the Process Step	Incident and patient reporting data are entered into the prehospital computerized reporting system.
Description	Description of the Process Step	
DateTimeCreated	Date and time row was created	
DateTimeUpdate	Date and time row was last updated.	
UpdatedBy	Who updated the row	

DEFINITION

Assigns a process step to a process noting the date and person making the assignment.

TABLE

HAS		
FIELD	DESCRIPTION	EXAMPLE
HasID	Unique Identifier	
ProcessID	References PROCESS	
ProcessStepID	References PROCESS STEP	
Sequence Number	A number designating the order in which the process step occurs.	1
DateTimeEffective	The date and time the process step was implemented	May 31, 1996
DateTimeTerminated	The date and time the process step was no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage

PROCESS_STEP_ATTRIBUTE

DEFINITION

An attribute of a process step. An attribute is anything about a process step that adds to the understanding of process and the effects on the data elements.

TABLE

PROCESS_STEP_ATTRIBUTES		
FIELD	DESCRIPTION	EXAMPLE
ProcessStepAttributeID	Unique Identifier	
Name	Name given to the process attribute	Report Audit Trail
Description	Description of the Process Attribute	The software allows changes to patient reports by authors and designated individuals with special privileges. The system does not keep track of all report changes; however, it does keep track of all report status changes, who makes them, and the time they were made.
DateTimeCreated	Date and time row was created	

PROCESS_STEP_ATTRIBUTES		
FIELD	DESCRIPTION	EXAMPLE
DateTimeUpdate	Date and time row was last updated.	
UpdatedBy	Who updated the row	

(PROCESS_HAS_PROCESS_STEP) _CONTAINS_PROCESS_STEP_ATTRIBUTE

DEFINITION

Assigns a process step attribute to a particular process/process-step combination.

TABLE

CONTAINS		
FIELD	DESCRIPTION	EXAMPLE
ContainsID	Unique Identifier	
HasID	References HAS	
ProcessStepAttributeID	References PROCESS STEP ATTRIBUTE	
DateTimeEffective	The date and time the assignment is valid	May 31, 1996
DateTimeTerminated	The date and time the assignment is no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage

ANOMALY

DEFINITION

An anomaly type found as a result of the process step attribute attribute. An anomaly type found in an element.

TABLE

ANOMALIES		
FIELD	DESCRIPTION	EXAMPLE
AnomalyID	Unique Identifier	
Name	Name given to the anomaly	Missing Values Unknown
Description	Description of the Anomaly Type	For some reason that is unknown values are missing from this element.
DateTimeCreated	Date and time row was created	
DateTimeUpdate	Date and time row was last updated.	
UpdatedBy	Who updated the row	

((PROCESS_HAS_PROCESS_STEP) _CONTAINS_PROCESS_STEP_ATTRIBUTE) _ONE_OR_MANY_(PRODUCES_ANOMALIES)

DEFINITION

Identifies the one or many process attributes that produce an anomaly.

TABLE

ONE_OR_MANY		
FIELD	**DESCRIPTION**	**EXAMPLE**
ContainsID	References CONTAINS	
ProducesID	References PRODUCES	
DateTimeEffective	The date and time the assignment is valid	May 31, 1996
DateTimeTerminated	The date and time the assignment is no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage

(((PROCESS_HAS_PROCESS_STEP) _CONTAINS_PROCESS_STEP_ATTRIBUTE)_ONE_OR_MANY) _PRODUCES_ANOMALIES

DEFINITION

An anomaly produced by one or more process step attributes.

TABLE

PRODUCES		
FIELD	**DESCRIPTION**	**EXAMPLE**
ProducesID	Unique Identifier	
AnomalyID	References ANOMALY	
Description	Description of the Anomaly	In the prehospital reporting system many screens contain fields which do not require the entry of a value. It was found that fields of this sort often did not contain values even though a value would have been applicable.

PRODUCES		
FIELD	DESCRIPTION	EXAMPLE
MeasuredYN	Was the anomaly measured (e.g. percentage of affected fields calculated). If no, indicates that the anomaly has been found to exist, but was either not measured or could not be measured.	No
MeasureDescription	Description of the measurement taken.	
DateTimeEffective	The date and time the assignment is valid	May 31, 1996
DateTimeTerminated	The date and time the assignment is no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage

DATABASE

DEFINITION

A collection of documents or records.

TABLE

DATABASE		
FIELD	DESCRIPTION	EXAMPLE
DatabaseID	Unique Identifier	
Name	Name given to the database	Prehospital Care Reports
Description	Description of the database	Set of relational tables making up a Prehospital care report in the prehospital information system.
System_Location	System database is maintained on (e.g. ABC Patient Care Software on UNIX server in XYZ outpatient clinic) or location of database (e.g. file cabinet in XYZ outpatient clinic)	Prehospital system with relational database located on UNIX system.
Contact	Who to contact for information on the database	
DateTimeCreated	Date and time row was created	
DateTimeUpdate	Date and time row was last updated.	
UpdatedBy	Who updated the row	

DOCUMENTS_OR_RECORDS

DEFINITION

A document or electronic record consisting of fields for capturing data.

TABLE

DOCUMENTS_OR_RECORDS		
FIELD	DESCRIPTION	EXAMPLE
DocRecID	Unique Identifier	
Name	Name given to the document or record (table)	Prehospital Care Main Patient Table
Description	Description of the document or record (table)	Table name rnraemssss. A relational table.
DateTimeCreated	Date and time row was created	
DateTimeUpdate	Date and time row was last updated.	
UpdatedBy	Who updated the row	

DATABASE_CONTAINS_DOCUMENT_OR_RECORD

DEFINITION

Assigns a document or record to a database.

TABLE

CONTAINS		
FIELD	DESCRIPTION	EXAMPLE
ContainsID	Unique Identifier	
DatabaseID	References DATABASE	
DocRecID	References DOCUMENT OR RECORD	
DateTimeEffective	The date and time the process step was implemented	May 31, 1996
DateTimeTerminated	The date and time the process step was no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage

ELEMENT

DEFINITION

Element is a single datum or field existing in a record or document.

TABLE

Element		
Field	**Description**	**Example**
ElementID	Unique Identifier	
Name	Name given to the a physical element (e.g. the field name in the table or label on a form)	trauma_yn; situation_found_code
Definition	Definition of the element	Flag for identifying whether a patient was entered into the trauma system. The code for the situation found at the scene.
Comments		
DateTimeCreated	Date and time row was created	
DateTimeUpdate	Date and time row was last updated.	
UpdatedBy	Who updated the row	

(DATABASE_CONTAINS_DOCUMENT_OR_RECORD) _CONTAINS_ELEMENT

DEFINITION

Assigns an element to a database/record relation.

TABLE

Contains		
Field	**Description**	**Example**
ContainsID	References CONTAINS from the (Database_contains_Document_or _Record) relation	
ElementID	References ELEMENT	
DateTimeEffective	The date and time the process step was implemented	May 31, 1996
DateTimeTerminated	The date and time the process step was no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage

ELEMENT_ATTRIBUTE

DEFINITION

An attribute of an element. An attribute is anything about the element that adds to the understanding of the element. Ideally anything desired to be known about an element would be stored here; however, it may be preferable to keep some standard attributes in the element table such as: data type, length, description.

TABLE

ELEMENT_ATTRIBUTE		
FIELD	**DESCRIPTION**	**EXAMPLE**
ElementAttributeID	Unique Identifier	
Name	Name given to the attribute	Default Value; Data Type; Length; Allows Nulls; Allowed Values
Description	Description of the attribute	The list of allowed values for the element.
DateTimeCreated	Date and time row was created	
DateTimeUpdate	Date and time row was last updated.	
UpdatedBy	Who updated the row	

((DATABASE_CONTAINS_DOCUMENT_OR_RECORD) _CONTAINS_ELEMENT)_CONTAINS_ELEMENT_ATTRIBUTE

DEFINITION

Assigns an element attribute to an element.

TABLE

CONTAINS		
FIELD	**DESCRIPTION**	**EXAMPLE**
ContainsID	References CONTAINS from the ((Database_contains_ Document_or_Record) _contains_element) relation	
ElementAttributeID	References ELEMENT ATTRIBUTE	
Value	The value for the attribute. For example, if a default attribute is designated what the default value is.	*(List of allowable codes)* CA – cardiac arrest TA – traffic accident BU – burns
Attachment	It may be easier to attach supporting documents in case where the value may contain many items such as set of allowable codes	*(file location)*

CONTAINS		
FIELD	DESCRIPTION	EXAMPLE
DateTimeEffective	The date and time the process step was implemented	May 31, 1996
DateTimeTerminated	The date and time the process step was no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage

((DATABASE_CONTAINS_DOCUMENT_OR_RECORD) _CONTAINS_ELEMENT) _ONE_OR_MANY_SUFFERS_FROM_ANOMALY

DEFINITION

Identifies which element or group of elements suffer from an anomaly.

TABLE

ONE_OR_MANY		
FIELD	DESCRIPTION	EXAMPLE
ContainsID	References CONTAINS from the ((Database_contains_ Document_or_Record) _contains_element) relation	
SuffersID	References SUFFERS_FROM	
DateTimeCreated	Date and time row was created	
DateTimeUpdate	Date and time row was last updated.	
UpdatedBy	Who updated the row	

((DATABASE_CONTAINS_DOCUMENT_OR_RECORD) _CONTAINS_ELEMENT) _ONE_OR_MANY_SUFFERS_FROM_ANOMALY

DEFINITION

An anomaly found one or many elements.

TABLE

SUFFERS_FROM		
FIELD	DESCRIPTION	EXAMPLE
SuffersID	Unique Identifier	
AnomalyID	References ANOMALY	
Description	Description of the Anomaly	The majority of records do not have a value in patient condition at transport.

SUFFERS_FROM		
FIELD	DESCRIPTION	EXAMPLE
MeasuredYN	Was the anomaly measured (e.g. percentage of affected fields calculated). If no, indicates that the anomaly has been found to exist, but was either not measured or could not be measured.	Yes
MeasureDescription	Description of the measurement taken.	For a two month period only 481 out of 1358 records (35%) had values for patient condition at the time of transport.
DateTimeEffective	The date and time the assignment is valid	May 31, 1996
DateTimeTerminated	The date and time the assignment is no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage

(...PRODUCES_ANOMALY)_IDENTIFIED_WITH_ (...SUFFERS_FROM_ANOMALY)

DEFINITION

Shows that an anomaly that was produced from process attributes can be identified with an anomaly found in an element or group of elements. This table is necessary because not all anomalies found in a dataset will be able to be traced back to specific process attributes. Likewise a process attribute may seem to produce anomalies that cannot be found in a specific dataset.

TABLE

IDENTIFIED_WITH		
FIELD	DESCRIPTION	EXAMPLE
ProducesID	References PRODUCES	
SuffersID	References SUFFERS_FROM	
DateTimeCreated	Date and time row was created	
DateTimeUpdate	Date and time row was last updated.	
UpdatedBy	Who updated the row	

(PROCESS_HAS_PROCESS_STEP)_GENERATES_ (((DATABASE_CONTAINS_DOCUMENT_OR_RECORD) _CONTAINS_ELEMENT)

DEFINITION

Shows the relationship between process and elements. Process generates elements. Keeping this level of detail for every step of a process may be more than those using the repository may want to record. At the minimum the final dataset resulting from a process should be recorded. The rest may be optional.

TABLE

GENERATES		
FIELD	**DESCRIPTION**	**EXAMPLE**
GeneratesID	Unique Identifier	
HasID	References CONTAINS	
ContainsID	References CONTAINS from the ((Database_contains_ Document_or_Record) _contains_element) relation	
DateTimeEffective	The date and time the assignment is valid	May 31, 1996
DateTimeTerminated	The date and time the assignment is no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage

PROCESS_HAS_PROCESS_STEP)_INPUTS_INTO_(((DATABASE_CONTAINS_DOCUMENT_OR_RECORD) _CONTAINS_ELEMENT)

DEFINITION

Shows the relationship between process and elements. Elements become input into the process. Keeping this level of detail may be more than those using the repository may want to record. This table should be optional.

TABLE

INPUTS_INTO		
FIELD	**DESCRIPTION**	**EXAMPLE**
HasID	References CONTAINS	
ContainsID	References CONTAINS from the ((Database_contains_ Document_or_Record) _contains_element) relation	
DateTimeEffective	The date and time the assignment is valid	May 31, 1996

Inputs_Into		
Field	Description	Example
DateTimeTerminated	The date and time the assignment is no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage

PROCESS_STEP_ATTRIBUTE_ INVOLVES_ELEMENTS

DEFINITION

The elements involved with a process attribute. This implementation follows the design in Figure 6.

TABLE

Involves		
Field	Description	Example
ContainsID	References CONTAINS	
GeneratesID	References GENERATES	
DateTimeEffective	The date and time the assignment is valid	May 31, 1996
DateTimeTerminated	The date and time the assignment is no longer valid	
DateTimeCreated	Date and time row was created	
DateTimeUpdated	The date and time this row updated	Oct 23, 1998
UpdatedBy	Who updated the row	Annette Savage