

DESIGN OF A LARGE DATA BASE  
A METHODOLOGY COMPARISON *meo*

by

JAMES R. WILSON

B.S., Brigham Young University, 1970

-----  
A MASTER'S REPORT

Submitted in partial fulfillment of the

requirements of the degree

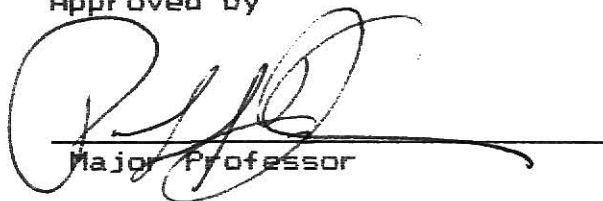
MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY  
Manhattan, Kansas

1983

Approved by

  
Major Professor

LD  
2668  
R4  
1983  
W54  
c. 2

#### ACKNOWLEDGEMENTS

I would like to thank Dr. Paul Fisher for his tutelage and assistance during the project and this report.

I would also like to thank my lovely wife, Christel, for her support and the time she sacrificed to allow the completion of this report.

**ILLEGIBLE**

**THE FOLLOWING  
DOCUMENT (S) IS  
ILLEGIBLE DUE  
TO THE  
PRINTING ON  
THE ORIGINAL  
BEING CUT OFF**

**ILLEGIBLE**

## TABLE OF CONTENTS

Introduction.....	1
Chapter 1. Current Design Methodologies.....	3
Chapter 2. Background and Analysis Phase.....	14
Chapter 3. Methodology 1: E-R Model Generation.....	26
Chapter 4. Methodology 2: Document Handler.....	34
Chapter 5. Comparison and Analysis.....	41
Chapter 6. Conclusion.....	53
Appendix A. Sample listing of Procurement Documents.....	A-1
Appendix B. Document Entity Diagrams.....	B-1
Appendix C. Operational Listing of Procurement Documents....	C-1
Appendix D. Data Structure of Ozark Forest Data Base.....	D-1
Appendix E. Data Model of Ozark Forest Data Base.....	E-1

## INTRODUCTION

The problems associated with the design of a large data base are often complex, time-consuming and not easily resolved by current technology. Much effort in terms of research, papers, publications and even textbooks has been expended in an attempt to resolve this problem. However, the basic problem still remains, how does a database designer effectively collect, examine and structure information from an organization into a system which will support current database technology, meet the goals of the users and yet build in sufficient flexibility to meet any future evolution of the organization's information needs?

This report does not propose a solution to those problems but rather looks at the development of a large data base being currently designed by NDX Systems Corporation for the United States Forest Service. It will examine the approach used to collect, examine and organize the information into a data structure which can be implemented on the systems available to the Forest Service. It will also report on two methodologies used by the author to design one section of the database and compare their effectiveness as well as any problems noted in their use. It will analyze the manual and automated methodologies utilized in the design process and recommend from the lessons learned in the project some steps which might be used

to facilitate future design efforts.

Chapter 1 of the report will briefly examine some of the current methodologies for logical data base design as well as some of the automated tools currently available to assist the designer in his task.

Chapter 2 will provide a summary of the background of the design project and discuss the analysis phase of the design process.

Chapter 3 will discuss the E-R Model Generation methodology.

Chapter 4 will discuss the Document Handler methodology.

Chapter 5 will compare the two methodologies, discuss problems encountered during the design process and examine these methodologies as to their applicability in future design efforts.

Chapter 6 will be a brief conclusion of the report.

## CHAPTER 1

### CURRENT DESIGN METHODOLOGIES

The project of designing a large data base is one which should not be initiated by someone with little design experience or someone who is not willing to expend considerable time and effort in its completion. If anxiety and frustration cannot be tolerated then the process required to arrive at an effective design should be left to others willing to assume these problems. It is not that the actual design of a database is an impossible task, nor is it the fact that there are no means with which to perform the process. It is, rather, the proliferation of ideas and methodologies which makes the design of a database a difficult task.

Database design is defined to be the process of developing a database structure from user requirements. [TEOR 82] Although the definition seems simple enough, the actual process is not. This may be witnessed by the fact that the approach or approaches to be used are not as clearly understood. Some practitioners have argued that there are at least two separate steps in the design process: the design of a logical database structure, describing the user's view of the data, which is processible by a data base management system and the selection of a physical structure that includes data representation or encoding, access methods, and physical organization of data.

Teorey and Frey [TEOR 82] list two phases also but they are not the same as those already discussed. Their phases are first, the analysis and design phase which consists of a requirements formulation and analysis step, a conceptual design step, and implementation design step and a physical design step. Second, the database implementation and operation phase which consists of the database implementation, operation and monitoring and modification and adaptation steps. In his text, Cardenas [CARD 79] lists three phases: the logical design phase, the physical design phase and the data base loading and operation phase. Still others list four and some even five phases to the process. It can be seen then that other than the logical/physical delineation, the overall structure of the process has yet to be defined. For the purpose of this report, the process will be considered to be a four phase process beginning with the analysis phase and followed by the logical database design phase, the physical database design phase and the implementation phase.

The analysis phase is perhaps the most ill-defined, difficult and time-consuming phase of the design process. It is, however, the first and most important because the output of this phase will determine the course of the remaining three phases. The major requirement of this phase is to collect information about the data content and processing requirements from the potential users of the database and to determine as much as possible the flow of information, volume of usage, and data requirements of the user organization. The organizational objectives, derivation of specific database requirements from the objectives or the

management and other user personnel, and documentation of those requirements in a form agreeable to both user and designer should be achieved.

The objectives of this phase cannot be reached by any well-established, proven process. There are many approaches including the conduct of a lengthy analysis of the organizational system by a team of systems analysts, interviewing management and nonmanagement personnel to determine requirements, collecting all the organizational documents as a basis for determining the data requirements or a combination of two or more of these or other processes.

Some of the necessary information requirements to be collected should include constraints such as security, system availability, reliability and primary business functions. This phase is the source of concern among designers as they attempt to resolve the problem of how best to conduct the collection process without unduly disrupting the functions of the organization for which the design is intended. This process will be discussed in more detail when examining the example later in the report.

The next phase is the logical design phase which concerns itself with the description and synthesis of diverse users' information requirements into a preliminary design. This phase, like the first has no set methodology but rather adds to the designer's difficulties by offering several approaches to the conduct of this phase. The output of this phase is normally a data diagram which allows both the designer and the user to see the data items, their relationships and how the organization's data

requirements are to be handled. There are, as stated, several approaches to arriving at such a diagram and which will be used in determining the actual physical structure of the data be it network, hierarchical or relational. Some of the more prevalent methodologies will be briefly discussed at this point.

The evolution of logical design methodologies is one which has produced some interesting techniques. Early efforts such as the relationship matrix and frequency distribution diagram described in the Auerbach series [AUER 76] recognized the need to define the relationship among data elements. The matrix records the number of times a particular data element is associated with another element to support an elementary function. The frequency distribution diagram was used in conjunction with the matrix by creating the diagram of frequency of data element usage across each task. A data element is determined to be a key if its task usage falls in the high range or right tail of the distribution. A data element is an attribute owned by one key and referenced by many keys if it falls in the average range. If it falls in the left tail of the distribution then it will be an attribute referenced and owned by only one key. This type of information is extremely useful in determining the location of data within the data base organization.

E. F. Codd, credited with being the father of the relational model, proposed in the early 70's a design methodology based on a process of decomposition of large relations into smaller, normalized relations which could supposedly eliminate problems such as insertion, update and deletion anomalies in large data

bases. Normalization, the name given to the process, sought to insure the simplicity or atomic nature of database attributes and to eliminate partial and transitive dependencies among attributes and their keys. This methodology utilizes an already existing flat file or relational design and decomposes it, according to the functional dependencies among the data elements while maintaining the data content. Its goal was to make the database easier to understand and control, simpler to operate upon and more informative to the user [Codd 76]. This methodology has been the basis for other works and has been improved by Codd and others.

Peter Chen introduced the entity-relationship approach in 1976 [CHEN 76]. This approach attempts to incorporate the semantic meaning of data into a real world diagrammatic model. Chen defines objects of interest to the user as entities, the properties of those entities as attributes and the association among entities and attributes as relationships. The key to this approach is the adding of an intermediate stage in the logical design process. The designer first identifies the entities and relationships which are of interest to the organization and then views the data from the point of view of the whole organization which is independent of performance and storage considerations. This approach uses an easily understood diagram which illustrates the entities, attributes and relationships. It has the advantage of keeping the design process free of the database organization and is easily understood.

A functional decomposition approach proposed by Finneran and

Henry [FINN 77] is a top-down structured analysis of the elementary functions identified in the design analysis phase. The elementary functions, those performed by an individual or group of individuals, are identified by each business function being "decomposed" in a recursive process. Each function is represented in a functional tree and identified by a box with an action verb and object. The designer, working closely with the user, defines each element in the tree giving it an element name, brief content description, storage type and size. Once defined, a data element list is created and elements with duplicate names are resolved and the data structure is completed. This approach claims flexibility as its best asset allowing for changes in the organization's data processing requirements to be easily incorporated.

An entirely different methodology was proposed by Phillip Bernstein [BERN 76] to produce a third normal form relational data base. Bernstein's synthesis approach or Bernstein's algorithm, as it has been called, incorporates the Armstrong axioms for functional dependencies and develops a normalized relational database from the data elements and functional dependencies among those data elements. These data elements and dependencies are determined through the analysis of the organization's data. Briefly, the algorithm eliminates extraneous attributes by examining the closure property of the set of functional dependencies, determining a nonredundant (minimal) covering by eliminating those functional dependencies which can be removed without affecting the closure, partitioning

the functional dependencies according to common left hand sides, merging equivalent keys by examining possible bijections, eliminating transitive dependencies and then constructing the relations which consist of all the attributes appearing within a partition. The keys are determined to be the left hand sides of the remaining functional dependencies. This approach has the advantage of starting with only the elements of data and arriving at a relational model which can be easily implemented, easily understood and is normalized.

Furthering the work of E. F. Codd and alleging to incorporate the synthesis approach of Bernstein, Ronald Fagin [FAGIN 77] proposed his fourth normal form decomposition methodology. This methodology incorporates all the attributes found in an organization's data into one large and often unmanageable relation. Utilizing the functional and multivalued dependencies among the organization's data items. This large relation is then decomposed into a family of relational schemata that Fagin claims to be in fourth normal form. Roughly speaking, a relation is in fourth normal form if all dependencies, either functional or multivalued, are the result of keys. This approach has been proposed to be strictly stronger than Codd's third normal form decomposition approach. It handles multivalued dependencies and functional dependencies as a subclass of multivalued dependencies. It does, however, require the user to have an intuitive understanding of the decomposition process and the application of Armstrong's axioms to arrive at the final design. In their text book, Teorey and Frey [TEOR 82] propose two

additional approaches which they call entity analysis and attribute synthesis. Entity analysis is a top-down approach which divides the design process into four stages: view modeling, consolidation of views, schema mapping and physical design. The view modeling stage attempts to model the data required for the data base as visualized through the various perspectives of the organization. These are, according to the authors, four types of views, the corporate enterprise view, the application view, the information view, and the event view. The information represented in each of these views include the organizational objectives/constraints, processing information, information structures and relationships, and events and scheduling. During this stage the actual gathering of information from the various levels and departments of the organization occurs. Inputs from the executive, managerial and end-user levels are then consolidated into a single conceptual view represented as a high-level structure diagram. This diagram forms the foundation of the data base management approach and is the most important part of the design process. The basic constructs of the diagram are similar to those utilized by Chen in his E-R model. The process of consolidation begins with the information perspective to evolve the logical database design.

Attribute synthesis is a bottom-up approach as it begins with the low level data attributes from which higher-level entities and relationships are formed. This approach is also broken down into four stages: classification of attributes, composition of entities, formulation of relationships and graphical

representation [TEOR 82].

During the classification phase, a list of data elements generated by the various tasks of an organization is completed. These elements are then classified into one of two classes of entities: unique or nonunique. A unique entity is a data element which identifies a distinct or particular object. Nonunique entities exist when two or more unique entities are used to identify a set of data elements. The actual classification process uses a heuristic approach similar to the Auerbach matrix and frequency distribution discussed earlier.

The formulation of the data relationships in either of three types - between entities, between entities and attributes, and between attributes - evolves by matching the policy statements obtained during the management interviews with the list of entities and attributes. Once this formulation is complete, the entity-relationship model is used to graphically portray the full information structure.

The structure is then interpreted so that it may be verified by all users. This interpretation process involves stating dependencies, determined by examining the structure diagrams, defining the implication of each dependency, defining what information will be lost if an entity is removed, and defining those entities where no dependencies exist. The interpreted structure is then examined for implications of possible changes and presented to the user for verification.

As can be seen by the brief discussion of these various methodologies, similarities exist in some of the aspects but

terminology and process steps do not always provide the designer with an easy choice of which methodology will best suit his particular design problem. It is the goal of these methodologies to eventually provide an automated approach to database design which will produce a useful structure in a reasonable time with a reasonable amount of effort. Some steps have been taken toward this goal such as IBM's Database Design Aid (DBDA) described in Hubbard and Raver [HUBB 75] which clusters the data elements in accordance with the associations in such a way that each cluster contains exactly one key. Associations between keys become associations between clusters. These clusters are then implemented as record types and the association as set types in the CODASYL model.

The Problem Statement Language/Problem Statement Analyzer (PSL/PSA) [HERS 75] is another computerized approach which allows the user to describe the problem in PSL as a collection of objects within the proposed database system, attributes of these objects, and relationships between these objects. Descriptions of these objects are also accepted and stored as part of their definitions, along with more detailed information concerning data for each entity and relationship. Usage specifications may be made in terms of PROCESS, INPUT, OUTPUT and other types of objects. The PSL/PSA system stores user-supplied system description information in a database. This information can then be checked for consistency and used for printing out various documents and reports from the stored system description which will be utilized to assist the designer in his efforts.

Another system called CASCADE [AANS 72] is used to perform a bookkeeping function for system requirements. Information is stored as a collection of related permanent information, permanent information sets and permanent message objects. This information is then used in the same manner as the system description information in PSL/PSA.

Mier Cohen [COHEN 81] in his master's report at Kansas State University proposed a system called Document Handler which uses the information found on an organization's documents as the basis for the design process. This system allows the user to enter the information found on a user document and determine a key or identifier for each document. The system then will provide the user with a listing of the documents in the organizational system, the columns or attributes in the system which can be checked for uniqueness, a listing of documents with their associated columns or attributes, a column or attribute cross reference, and if requested will prepare the information for application by Bernsteins' algorithm by determining functional dependencies. This system will be examined more closely later in this report as it was utilized in the actual design problem.

These few automated systems each perform a valuable function but do not provide the designer with a tool which can be utilized from the beginning of a design process through to the end. It is for this reason that the design of a large database is often a frustrating and generally a time-consuming process.

## CHAPTER 2

### BACKGROUND AND ANALYSIS PHASE

The use of computers in the National Forest Service has received considerable attention in the past few months due to a large national purchase of computers for all of the national forests. This purchase was due in large part to several problems in the Forest Service not the least of which was the turn-around time needed to get data in and out on several of the existing programs. Although computers have been a part of the Forest Service for years, their use has not been consistent. With an organization as large in geographical area and in volume of data used, automation of many of the processes was long overdue. It is expected that it will take about five years for the Forest Service to integrate the new machines into their system and for the personnel to convert to the system and accept the benefits of an automated process. Some National Forests decided to immediately examine how these new systems could be used to help solve the information flow and maintenance difficulties that had been experienced under the old system.

The Ozark/St. Francis National Forest was one of those forests which recognized the importance of managing the data resources in a more efficient manner. They also recognized that several problems existed at the local level which included the need for common software for various uses, converting programs

previously used at a centralized location for application at the local offices, and training within the Forest Service on the use of the systems and the capabilities to write application software. Additionally, it was recognized that no national data base management system has been devised to be used on the newly purchased systems at the local level nor had any plans been made on how to integrate the district machines within the already existing National Forest machines and databases.

It was at this point that the NDX Systems Corporation became involved in the process. In June 1983, representatives from NDX went to Russellville, Arkansas to collect the information needed to describe the data processing activities of the Ozark National Forest. During this period the documents which were normally used in the Forest Service Supervisor's Office were collected and interviews of forty-four employees were conducted to determine the functional areas, information flow and usage factors of the data resources of the organization. This process was done without much disruption to the normal business functions of the Supervisor's Office.

A total of 715 documents were collected in three primary application categories: Budget and Finance, Personnel and Forest Resources. It was determined through the interview process that several smaller functional areas exist within each of these categories.

The purpose of this portion of the analysis phase was to associate the data normally used within the office with its flow throughout the organization and thereby determining their usage

patterns and necessary data items. In all of the design methodologies briefly discussed in chapter 1, this step is essential if a database design is to be accomplished which will accommodate the application functions of the organization within the dataflow structure.

During these initial steps in the analysis phase, the primary goal was to categorize the documents as well as the information collected. As is pointed out in a recent paper [FISH 81], the design of a database requires the consideration of a great deal of information. Since this information and its function within the organization will determine its significance in the database and affect the overall design, information is categorized into two groupings. The primary grouping contains data element names, data element size, data element value domains and data element usage. These items will make up the body of the database whereas the secondary grouping which consists of various usage views and their associated access paths, alterability of data values, difference between representations and associated pseudonyms are necessary to arrive at the physical structure of the database.

It was with these considerations in mind that the documents were collected and collated into one of three categories: input, output or resident. In this same paper [FISH 81] it was stated that within an organization such as the Forest Service, it is clear to the individuals providing the information exactly which documents are 'input' and which are 'output'. It is not as easy, however, to delineate those documents which deal with the data

items and form the basis of an organization's business functions and as such become the basis for the desired database elements. In order to alleviate the difficulty in collecting and identifying the 'resident' documents, careful attention was paid to the existing information structure in the Forest Supervisor's office, the interrelationship between the activities using a particular document and usage factors of the documents and their data items. An integrated approach which utilized aspects of the Auerbach, Finneran and Henry and Teorey and Frey approaches earlier discussed, was used to accomplish this document identification task. Within the Forest Service, such things as recreation areas, timber resources by unit, personnel, funds and other such elements were determined to make up the resident elements.

The identification of the input and output documents was a much easier process since they serve as the external manifestations of the organization. Utilizing these documents provides a means of indirectly determining the contents of the database. The methodology used to determine the resident documents will be discussed in detail in Chapters 3 and 4 of this report.

As stated earlier, the initial number of documents collected from the Forest Service totaled 715. This number included documents which were stored but not used and excluded documents which were local to a particular departmental function and were not stored in a common location. The process of interviewing personnel within each major Forest Service group or function

aided in the elimination of over 200 documents as obsolete and also resulted in the addition of some of the local documents for consideration in the overall database design. The number of forms which remained was reduced to 455.

It was also determined in this process that the documents necessary to comprise the actual database were not present. It was also discovered that many forms come into the Supervisor's Office and were simply filled out for later reference and as such had to be considered as part of the actual resident documents of interest. Another problem, which was expected, was that the actual information kept on many of the forms is not unique and would cause duplication in the database or require different referencing methodologies. Furthermore, several forms within the Supervisor's Office exist to supply information to a larger, centralized computing resource. These systems address various local and national concerns and were given mixed evaluations by those interviewed.

Having completed the collection process, it was then necessary to identify the major divisions, the various subfunctions within each division and the number of documents by type found in these areas. Tables 1-3 identify these divisions and the number of documents found in each area.

TABLE 1 FORMS ANALYSIS  
BUDGET AND FINANCE

<u>FUNCTION</u>	<u>TYPE AND NUMBER OF FORMS</u>		
	<u>Input</u>	<u>Output</u>	<u>Resident</u>
Collection	-	16	-
Fleet Equipment	-	4	-
Utility and Supplies	-	2	-
Obligations	-	8	-
Travel	-	10	-
Petty Cash	1	-	-
Collections	-	-	4
Collections, Disbursements	3	-	-
Budget Planning	-	6	-
Personnel Claim	5	-	-
Equipment Control	4	-	-
Financial Planning	9	-	-
Public Information Office	-	-	3
Property and Equipment	-	2	-
Contracts	-	7	-
Quarters and Space	-	9	-
Purchasing	-	13	-
Equipment and Operation	4	-	-
Building and Property	4	-	-
Acquisition of Equipment and Supplies	9	-	-
Building and Property	19	-	-
Bids and Contracts	-	-	13
<u>TOTAL</u>	<u>58</u>	<u>77</u>	<u>20</u>

TABLE 2 FORMS ANALYSIS  
PERSONNEL

<u>FUNCTION</u>	<u>TYPE AND NUMBER OF FORMS</u>		
	<u>Input</u>	<u>Output</u>	<u>Resident</u>
Evaluation and Hiring	-	-	9
Individual Personnel	-	-	4
Employee Training and Performance	-	7	-
Employee Finances	14	10	-
Employee Medical and Accident	11	11	-
Employee Qualifications	8	-	-
Employee Position Request	5	-	-
Employee Application	5	-	-
Employee Insurance, Disability, Death	16	-	-
Employee Driving Record	3	-	-
<u>TOTAL</u>	<u>62</u>	<u>35</u>	<u>13</u>

TABLE 3 FORMS ANALYSIS  
FOREST RESOURCES

FOREST RESOURCES

<u>FUNCTION</u>	<u>TYPE AND NUMBER OF FORMS</u>		
	<u>Input</u>	<u>Output</u>	<u>Resident</u>
Timber Sale Finance	4	-	-
Compartment Timber Tally	7	-	-
Timber Sale Documents	4	-	-
Pesticide	1	-	-
Timber Sale Collection	-	4	-
Timber Sale Contract	-	13	-
Progeny and Silverculture	-	6	-
Road Specification	-	6	-
Silverculture	1	-	-
<u>SUBTOTAL</u>	<u>17</u>	<u>29</u>	<u>-</u>

FIRE MANAGEMENT

<u>FUNCTION</u>	<u>TYPE AND NUMBER OF FORMS</u>		
	<u>Input</u>	<u>Output</u>	<u>Resident</u>
Prescribed Burning and Fire	4	-	-
Fire Equipment and Manning	8	-	-
Fire and Weather	-	-	3
Fire Personnel	-	7	-
Manpower and Fire Equipment	-	-	8
Fire Report	-	5	-
Fire and Weather Status	-	4	-
<u>SUBTOTAL</u>	<u>12</u>	<u>16</u>	<u>11</u>

PUBLIC INFORMATION OFFICE

<u>FUNCTION</u>	<u>TYPE AND NUMBER OF FORMS</u>		
	<u>Input</u>	<u>Output</u>	<u>Resident</u>
Public Information	-	3	-
<u>SUBTOTAL</u>	<u>0</u>	<u>3</u>	<u>0</u>

LANDS

<u>FUNCTION</u>	<u>TYPE AND NUMBER OF FORMS</u>		
	<u>Input</u>	<u>Output</u>	<u>Resident</u>
System Plan	-	-	1
Land Exchange	-	2	-
Special Permit	8	-	-
Electronic Permit	4	-	-
Land Use Report	1	-	-
<u>SUBTOTAL</u>	<u>13</u>	<u>2</u>	<u>1</u>

WILDLIFE RANGE

<u>FUNCTION</u>	<u>TYPE AND NUMBER OF FORMS</u>		
	<u>Input</u>	<u>Output</u>	<u>Resident</u>
Permitted Land	-	-	1
Eagle-Osprey	2	-	-
Livestock Improvement	-	4	-
Livestock Grazing	9	-	-
Grazing Permit	-	5	-
Wildlife, Range, Resources	6	-	-
<u>SUBTOTAL</u>	<u>17</u>	<u>9</u>	<u>1</u>
<u>TOTAL</u>	<u>59</u>	<u>59</u>	<u>13</u>

An essential part of the methodologies for database design was to identify the usage volume or frequency of use of the data. Tables 4-5 illustrate the frequency of use for the input and output documents respectively. Determined as a result of the interviews and analysis of the information flow within the Forest Service, this information will be utilized in determining the actual physical structure of the database. Locating more frequently used items at the front of the data structure will facilitate more rapid and easier access to more commonly needed data items. The result of this step also results in the development of the access strategy of the data.

TABLE 4

## INPUT DOCUMENT FREQUENCY

Number of Times Document is Needed per Year	Number of Input Documents
1	4
10	2
12	4
15	2
17	1
20	1
25	3
30	2
40	2
50	6
60	3
100	2
124	1
125	1
137	1
150	1
180	1
250	1
350	1
400	1
500	1
800	1
900	1
1000	1
1300	1
2000	1
TOTAL	
8596	46

TABLE 5  
OUTPUT DOCUMENT FREQUENCY

Number of Times Document is Needed per Year	Number of Output Documents
1	2
2	3
4	2
6	1
7	1
8	1
10	5
12	4
15	1
16	1
20	8
25	3
30	1
35	2
40	2
45	1
50	4
60	1
65	1
75	1
80	1
100	2
150	2
200	4
300	7
350	1
360	1
400	3
800	1
1162	1
1400	1
2000	1
3000	1
4000	1
TOTAL	72
14828	

Having completed the initial collection and collation of the Forest Service's documents, the next step was to organize the data within the three primary functional areas and provide the Forest Service with an initial view of the documents with their associated data items. It was decided to utilize Document Handler, described in Chapter 1, to facilitate this process. Document Handler provides an automated means of organizing the documents into document headings and data items which are taken from the document itself and are deemed to be important. The program then organizes this data and provides the user with a listing of the document names within that section of the database, a listing of the document names and data items or column names associated with each document name and a cross reference of all data items or column names and those documents upon which they commonly or singularly appear. (See APPENDIX A for a sample listing of the Document Handler.) This information can then be utilized to resolve ambiguities, insure functional area completeness and correctness, eliminate redundant data items or resolve duplicate names and eliminate unnecessary data from the data base.

Prior to entering the data into the Document Handler program, each document was carefully examined to derive those items of data which were deemed necessary to maintain the integrity of the document and which could be use to store necessary data items which would be entered from input documents as well as provide necessary output data for output documents. The initial printout containing this first view of the data was

then sent to the Ozark National Forest for their consideration and clarification of ambiguous data items. This information was returned and a second run of the document less the eliminated items and with the necessary modifications was returned again to the Forest Service. This process continued for approximately three iterations until both the Forest Service and designer personnel were satisfied that the data remaining was both necessary and no longer ambiguous.

The analysis phase as discussed thus far required only two visits from the NDX team to the Supervisor's Office. This approach caused only a minimal disruption of normal business processes of the Forest Service. It was, however, thorough enough to provide sufficient understanding between the design team and the prospective users that would allow further development in a timely manner. The documents which fell in one of the three functional areas already discussed were treated separately during the analysis phase to expedite the process. The logical design process then proceeded on the Budget and Finance section which had been the first to be returned with all ambiguities addressed and other questions resolved. The remainder of this report will only address the design process as it was implemented in this area.

## CHAPTER 3

### METHODOLOGY 1: E-R MODEL GENERATION

The logical data base design phase of the data base design process is one of synthesizing the collection and associations of data to satisfy the information storage, retrieval and reporting requirements of the using organization [CARD 79]. This process is usually an iterative process, often involving trial and error and usually without any clearly defined methodology to assist the designer in his efforts.

The design of the logical database for the Budget and Finance section of the Ozark National Forest database presented some unique problems. Included among these problems were the lack of a predefined data structure to be used, the data base management system to be used and the lack of experienced personnel at the Forest Supervisor's Office having specific knowledge of the new system and its characteristics. This lack of experience was a problem which could be ameliorated somewhat by the experience of the design personnel in the application area, but the inability to communicate because of this lack of experience remained.

With this lack of predefined guidance, any of the methodologies previously discussed could have been utilized to arrive at the schema to be used and the data model which

represented it. It was decided that for the purpose of this report to use and compare two methodologies previously proposed by two former students of this institution, and determine their applicability to future design efforts. This chapter will discuss the first of those methodologies which was proposed by Darrell Woelk in his Master's Report [WOEL 81].

This methodology proposed the utilization of user documents in an automated generation of a database schema. More specifically, the report proposes a system which will utilize user documents to generate a graphical Document E-R Diagram which the system will interactively manipulate to generate a true E-R Diagram. Although the system proposed would be an excellent design tool if it could be implemented, it currently has not been implemented on any existing system. For the purpose of this report, the actual generation of the Document E-R Diagram and the subsequent steps were simulated manually. This then immediately increased the amount of time utilized in the design process using this approach and could be questioned as inserting some degree of bias into the comparison of the two approaches. Nevertheless, the report will attempt to be just in its appraisal of both methodologies.

This methodology uses the following algorithm [WOEL 81]:

1. Data base designer surveys the organization and enters the data concerning Documents, Data Items, Value Sets, Specifications of Fields and other associated items into the system.
2. System generates a Document E-R Diagram.

3. Designer analyzes Data Item Names for duplication and ambiguity with aid of the system.

4. Designer enters a description of each Relationship among Document Entities.

5. Designer indicates the ownership of Data Items which are common to more than one document.

6. System makes deletions from Document Entity Data Item Lists for Data Items which are not owned by the Document Entity. Any Data Item deleted from a Document Entity Data Item List is also deleted from any other associated Relationship Common Data Item Lists.

7. System eliminates Document Entities and Relationships which have empty Data Item Lists and Common Data Item Lists, respectively.

8. Designer manipulates the resulting Entity-Relationship Diagram using the SPLIT, MERGE, SHIFT, MOVE, and CHANGE operations of the system.

9. Designer specifies the mapping (1:N, N:1, M:N) of the relationships among Entity sets.

10. Designer identifies the key attributes for all Entity Sets and Relationship Sets.

As earlier stated, this system was not machine implemented and it was necessary to modify some of the steps to be able to utilize this design approach. Specifically, due to the length of some of the Document Entity Data Item Lists and the complexity of some of the initial drawings, the listing of these data items was

not included but rather was maintained on a separate listing.

The Document Entity Data Item Lists are the lists which reflect the Data Item Names entered from the user documents. An initial examination of the Budget and Finance Documents immediately demonstrated the complexity of this approach since there were a total of 77 documents, some of which had more than 20 data items to be listed. It was decided to use this approach only on a subsection of the Budget and Finance section consisting of 12 documents which were categorized as procurement documents.

The first three steps of the algorithm were then combined and drawn on one diagram (Appendix B, Figures 1 thru 1g). This diagram was modified from the stated algorithm, in the manner discussed earlier, in an attempt to keep an already cluttered diagram readable and manageable. The separate listing was referred to as needed (Appendix C). Only those data items which were found on more than one document were listed on the diagram as were the Common Data Item Lists which contain the data items common to the two Document Entities connected by the relationship. The selection of common data items proved to be more difficult to do manually but with the ability of the proposed system to match data items between two documents, this would be a simple task. Steps 1 and 3 of the algorithm were accomplished during the analysis phase and it was not necessary for the purpose of this report to repeat them manually. Step 2 was the most difficult of the steps of the algorithm to complete. The manual drawing of this diagram took three attempts before a diagram of any degree of readability resulted. This was due

mainly to the fact that the commonality of data items in the first generation of this Document E-R Diagram was numerous, resulting in over 40 Relationships and the associated lines connecting the Document Entities. It was immediately apparent that the placing of a Document Entity was important as it might serve better in the center of the diagram as opposed to the outside due to the number of Relationships in which it was involved. An example of the importance of this placement is shown in Appendix B, Figure 1. If the Purchase\_Order Document had been placed at the edge of the diagram, the complexity of the diagram would increase and the intertwining of relationship lines would be almost impossible to draw or manage in later steps. Step 4 (Appendix B Figures 2 thru 2d) of the algorithm is seemingly a trivial task but care should be taken to select Relationship names which portray some meaning of the interrelationship between the Document Entities. This not only assists in the design effort but also in the later use by the organization since these names can make understanding the database implementation easier. These names will also serve as documentation in the final model.

Steps 5 and 6 were combined with step 4 on the second diagram (Appendix B Figures 2 thru 2d). These steps are important in the manner in which they are accomplished. The automation of this process would certainly reduce the difficulty encountered in attempting to complete these steps as it proved cumbersome to accomplish manually. The system must be able to overcome the complexity of the diagram which would relieve the

designer from the impossible task of having to keep track of which Document Entity was the owner and from which Document Entity Data Item Lists and Relationship Common Data Item Lists the attributes were to be deleted. Once the owner has been selected, the deletion of the owned data item must be carried through all associated Document Entities and each associated Relationship prior to preceding on to another data item, otherwise it would be easy for the system to omit the elimination from Relationships and Document Entities not directly linked to the owner Document Entity. The failure to do this would leave the diagram cluttered and defeat the purpose of these steps.

The remainder of the steps 7 through 10 were combined on the last diagram (Appendix B Figures 3 thru 3c). The result of step 7 reduced the diagram complexity significantly and the E-R Diagram which resulted was easily seen. Steps 6 and 7 are perhaps the strong points of this approach. The data model which results is one which has received considerable attention for its adaptability and its ability to show the semantic relationships of the data items and entities. If this approach could be implemented the resulting diagram would be a boon to the efforts to automate data base design. In this particular subsection, of the eleven original Document Entities, only one was removed through this process. More notable was the reduction of the number of the Relationships and their Common Data Item Lists. From an original number of forty-six such relationships only twelve remained after the deletion process was completed. This understandably reduced the clutter from the original diagram and

gave an easily read and easily understood model.

This approach, although lengthy when attempted manually did result in a data model which could easily be mapped into the selected data structure. The resulting Data Item Lists and Document Entities make up the database schema which could be modified into a relational, hierarchical or network structure.

This is not an approach which could be easily implemented nor would it be an approach with which a novice designer could begin a design process. It requires some intuitive knowledge on the part of the designer to be able to select those entities which would be better placed in the center of the diagram as opposed to those which would be better placed on the edges of the diagram. Although this seems like an unimportant consideration, this selection process proves to be the key to making the diagram readable and manageable. Additionally, the designer must be able to select the owners of the data items effectively since the selection process may in fact determine whether or not one or more of the Document Entities will be eliminated. In the process of using this methodology for the procurement documents it was discovered that if the owner of some of the data items had been the Multiuse\_Standard Document as opposed to the Purchase\_Order Document that the Document Entity Certificate\_In\_Lieu\_Of\_Lost\_GBL would not have been eliminated nor would several of the relationships which were eventually removed. Incorrect selection of a Document Entity as the owner could leave the diagram cluttered and reduce the effectiveness of this approach.

As a manual process it would be an impossible task to handle

a database of any size. The larger the proposed database, the more impossible the task becomes because of the complexity and immensity of the diagram and the ability of the designer to be able to maintain an accurate record of all Relationships, Data Item Lists and their subsequent manipulations.

The methodology does propose an interesting approach to the automation of the design efforts. The algorithm is easily followed and presents no difficulty in the actual execution of the steps. The result, as stated, is a data model which can be easily read and is widely accepted in the database design environment. The actual implementation of this approach is questionable as will be discussed further in Chapter 5.

## CHAPTER 4

### METHODOLOGY 2: DOCUMENT HANDLER

The second methodology as proposed by Mier Cohen in his Master's Report [COHE 79] concerned a system for the automatic generation of relational data bases. The report discussed the methodologies used in the analysis phase and the logical design phase. The name given to this automated approach was Document Handler and its use in this design project will be discussed in this chapter.

This approach also utilized the documents of an organization as the basis of the design process. As the documents were analyzed it was noted that they could be categorized in one of four related entities: Documents, Document-Attributes, Columns, and Column-Attributes. Each document has a set of columns and a set of document attributes. Each column is owned by one document although the same column might appear in several documents. As a result, a column is uniquely identified by its name and by the name of the document on which it occurs. A column might also have a set of column attributes and they are also uniquely identified by a unique name and by the name of the column with which they are associated.

Document Handler is designed for interactive work and allows the user to enter the documents and columns and their associated attributes and the system then organizes these inputs into

listings which can be used to manipulate the documents and columns. Finally, it will prepare them to be run in the Bernstein Algorithm Program which results in the generation of a third normal form relational schema. The commands and actual methodology of their use can be examined in the report [COHE 79].

When the documents from the Forest Service were first collected, Document Handler proved invaluable in the early analysis of the information gathered as was discussed in Chapter 2. Once the analysis phase was completed, the determination of the resident documents was the next step to be accomplished. The Budget and Finance Documents totaled 77 after the analysis phase and they were then broken down into five subsections: Equipment, Property, Funding, Procurement, and Contracting. Each of these subsections was then manipulated utilizing Document Handler's MERGE and DOCUMENT SET commands.

The process began by determining those documents which could be placed in one of the five subsections by analyzing the documents again and grouping them according to the functional use of each document. Once this had been accomplished, those documents which were determined to be resident documents during the analysis phase were again examined for commonality of functions and of data items. Where possible these documents were then combined or new documents created which would minimize duplication while maintaining data integrity and insure no loss of data content. Once these resident documents had been determined, they were merged into one resident document using the MERGE command. This particular command will combine two

documents into one of a different name and eliminates duplicated attributes.

Once this merging process has been completed, the documents which were determined to be input documents were examined and intersections were performed with the resident documents. This was accomplished through the Document Handler command: DOCUMENT SET. The input documents are compared using the following set relation:  $I - (I * R)$  where  $I$  represents the Input Documents,  $R$  represents Resident Documents,  $-$  represents the set difference and  $*$  the intersection of the sets. The system performs this intersection and the result is a list of data items or document/column attributes which will either be null or will contain names of the Document-Attributes, Columns or Column-Attributes. If the list is null, this means that all of the names contained in the input documents were found in the name domain of the resident documents. If the list is not null this indicates that some of the data items occur on the input list which are either superfluous, will be consumed or need to be carefully evaluated to determine their importance in the data base and whether or not they should be added to the list of resident data item names to insure completeness of the data content.

This process required several iterations within each subsection before listings resulting from the process were null. Several data items were found to be similar except for spellings, abbreviations, pseudonyms or were found to be ambiguous in their meaning or use. These data items were collected and their

semantics and use were discussed with the Forest Service Supervisor's Office to determine if they were the same as other already entered data names or if they were essential in maintaining data content.

This process was repeated for the output documents until the resulting listings were null. The data items which appeared in these listings were examined to determine if they could be derived from data items already in the data base resident documents, i.e., if they could be computed from already existing data. Data item names which appeared in the Output-(Resident \* Output) listings were often dates which could be output using system capabilities rather than maintaining them within the data base as virtual data items. Where these data items existed, they were eliminated from the listings, and the intersection process repeated. Since the output documents reflect the external manifestations of the organization it was necessary to insure that all the data item names were either eliminated in the above considerations or were included in already existing resident documents or new resident documents generated which would include all the data items necessary to generate the output data items used in the reports and output documents of the Forest Service. Where problems arose which required clarification concerning the semantics of a data item or its necessity, the Supervisor's Office was contacted to insure clarification occurred prior to the final design steps being completed.

Once the process had been completed for all the subsections the resulting documents were combined as the tentative database

organization. All variations, synonyms, abbreviations and pseudonyms were resolved, data which resulted was minimized within the capabilities of the system, and finally keys were determined for each resident document. Of the original 77 documents, only 48 remained as resident documents in the tentative schema. In the procurement area the number of documents which composed the resident documents was reduced from 12 to 6 documents. The reduction was a result of the system and the designer removing documents which contained data that could be obtained from other resident documents in the database or combining documents which had a common purpose into one document while not resulting in loss of data integrity or data content. This tentative database was then 'prepared' for input into the Bernstein Algorithm Program by using the Document Handler command: PREPARE. This command takes all the resident documents and prepares a "data dictionary" of all the data elements and also prepares a listing of the functional dependencies within the tentative schema. The data dictionary is not the complete data dictionary as is found in most databases but rather a modified listing of the data item names and a three letter alphanumeric item code used in the functional dependency listing and the Bernstein Algorithm Program. In order for the data to be input into the Bernstein program, however, the data dictionary had to be removed from the input file.

The Bernstein Algorithm Program produced a third normal form relational schema from the tentative database. The algorithm which has been implemented at this institution is as follows:

[BERN 76]

Let  $F$  denote the set of functional dependencies as determined by the Document Handler.

1. Eliminate extraneous attributes from the left hand side of each functional dependency in  $F$ , producing a set  $G$ . An attribute is determined to be extraneous if its elimination does not alter the closure of the set  $F$ .

2. Find a nonredundant covering (set  $H$ ) of the set  $G$ .

3. Partition  $H$  into groups such that all of the functional dependencies in each group have identical left hand sides.

4. Merge equivalent keys.

5. Eliminate all transitive dependencies.

6. Construct the relations.

This process synthesizes the functional dependencies into a relational schema that is in third normal form, i.e., all domains of the schema contains single values, all nonprime attributes are fully functionally dependent on all the keys of a particular relation, and none of the nonprime attributes are transitively dependent upon any key in the relation.

The schema which resulted from this process was then examined to determine if any keys were eliminated, if any of the resident documents (relations) were eliminated or if any of the data items were eliminated. Where this occurred an examination of the intended relationships was conducted to insure that the intent of the data organization was maintained and that the semantics of the data content was preserved. If these problems

surfaced, new keys were determined for the resident documents affected and the process repeated until the schema which resulted had resolved these problems.

The resulting schema was then used to finalize the resident document set of the database. This document set made up the records of the Budget and Finance Section of the database. A diagram (Appendix D Figures 1 thru 5) was then drawn to illustrate the data organization and utilized the usage factors to place the records as optimally as possible. From this diagram a network data model was constructed (Appendix E Figures 1 thru 5) to demonstrate the access strategy of the database. A network model was chosen as it was the most likely structure to be implemented when the database is actually implemented. A data dictionary was prepared and these documents were forwarded to the Forest Service for their comments. At the time of this report no comments have been returned.

## CHAPTER 5

### COMPARISON AND ANALYSIS

The use of the two methodologies as discussed in Chapters 3 and 4 was a time-consuming, detailed process. As can be expected initially in any design attempt, the amount of time spent learning the new systems amounted to considerably more time than that actually spent in the process of designing the database. The two methodologies each have some strong and weak points and this chapter will discuss these points plus problems encountered, make a comparison where appropriate and provide an analysis of the usefulness of either system in future design efforts.

Both of the methodologies utilize the user organization documents as the basis for the design of the actual database. Both designers realized that this was a logical starting point to determine the information requirements which would be supplied by the database. During the analysis phase, however, differences arose in the two methodologies concerning the importance of such factors as usage factors of the documents, involvement of the user organization in the resolution of ambiguities and other conflicts and grouping of documents in areas of functional commonality. The first approach discussed did not address the issue of usage factors as it relied more heavily on the relationships among the documents. The designer who uses this approach is not concerned with the usage factors even after the

E-R model is completed. This fails to consider the optimal access strategy of front-loading the database with those records whose frequency of use is the greatest. The efficiency of the resulting design is not one of the considerations that the author of this approach considered but if this methodology is to be of any value, the document usage factors must be incorporated into the final E-R model.

The second approach relied both on usage factors and the relationships among documents which were determined during the visits to the Ozark National Forest. It should be noted here, however, that the initial attempt at organizing the documents according to functional areas and the interrelationships was not very well done. The initial design steps of this methodology had to be repeated because of the failure to effectively carry out this important analysis step while interviewing the personnel and analyzing the information flow. A careful reexamination of all the documents had to be conducted and the documents regrouped before the design steps could be continued. The careful evaluation of each document and its use and flow within the organization is essential and must be an important part of the interviews with the organization personnel to insure that time and efforts are not wasted in later design steps which depend upon the correctness of this initial analysis.

The two approaches differed also in the involvement of organizational personnel in the resolution of ambiguities and conflicts. The first approach relied almost entirely upon the proposed system and the designer's intuition in determining the

resolution of these problems. The ability of the machine to make such judgments is unlikely and the burden then falls on the designer. The second approach relied more heavily on the constant interaction between the design team and the Ozark Forest personnel to resolve these problems. The difference of the two approaches in the degree of user involvement is then a question of whether or not the design team has enough knowledge and understanding of the data and its use to make such determinations and in so doing risk the probability of conflicts arising again after the design has been completed and implemented on the organization's system. If the organization is included in the resolution of these problems during the design process, the risks of problems arising later in the implementation and use of the database is decreased and the organization shares in any such risks as well as feeling a part of the design process.

Since the first approach was not implemented except for the manual simulation as discussed in Chapter 3, the actual value of this system during the analysis phase is uncertain. Document Handler proved to be a very valuable tool during this phase since it provided printouts which were well organized and easily understood. These printouts served as a means of communication between the organization and the design team. Such problems as semantics of data items, ambiguities, pseudonyms and deleting data items which were the same but had different spellings or slightly different names were easily resolved due to the listing and the cross-referencing capabilities of the system. Additionally, the listings as provided by Document Handler

assisted the designer in grouping the documents since data items which appeared on several documents inherently pointed out some relationship among the documents. Keys were also easier to determine with Document Handler, whereas the first approach relied on the designer to determine the keys.

The first approach depended upon the identification of common data items between Document Entities to establish Relationships and their Common Data Item Lists. This was initially attempted manually and proved to be a process which was lengthy and unmanageable even with only 12 documents. Document Handler was used to speed up the process and provided through its cross-reference listing, data items which in fact made up the Common Data Item Lists of each Relationship. Had the first approach been implemented this problem possibly would not have occurred but it is interesting to note that Document Handler could be integrated to provide the first system with both a means of entering input and a means of determining the Document Entity Data Item Lists and the Relationship Common Data Item Lists.

Once these Data Item Lists have been determined, the process of determining the owners of the individual data items becomes an important step in this algorithm. It is at this point that the implementation of this approach is highly questionable. With the best of the graphics monitors being considered and using high resolution, state-of-the-art graphics, the maximum number of lines of text (in this instance the Data Item Lists) which could be shown on the screen is approximately 24. The number of Relationships, as illustrated in the diagrams, which might

contain one or more of these data items in its Common Data Item List can range from one to many depending on the size of the database. The ability of a graphics monitor to provide a window which would be capable of showing more than one or two documents simultaneously or showing all of the relationships which have a particular data item in their Common Data Item List currently does not exist. The ability of a designer to keep track of these data items, which document is the owner and from which Document Entity a data item would be removed is made even more difficult by the inability of the monitor to provide a display large and encompassing enough to assist in this task. It is even doubtful that documents having more than 24 Document Entity Data Items could be effectively displayed. The process of determining the Document Entities which should be owners requires the designer to be able to see the entire diagram. He must be able to see how the data items which are owned by one Document Entity appear in other Relationships involving Document Entities which are not owners of that Data Item. The ability to see the entire diagram makes it possible to select the best owner rather than just the first entity seen which possesses a particular data item appearing in more than one Document Entity. If the designer is the one who has to make the decision on the owners of these data items then his task is extremely difficult if not impossible if he must rely on the current technology available on graphics monitors. Assuredly, if this step in the methodology could be successfully implemented it would reduce the complexity of this approach and make it a valuable design tool. It would not be a

panacea to the automation goals but it would be a large step forward.

The second approach does not rely on graphics but rather on easily read and easily produced listings. These listings provide the designer with an easily manageable media through which he can communicate the organization of data, keys, functional groupings and data item ambiguities and conflicts. Because of the cross-referencing capabilities of the system, no graphics are needed to see the relationships between documents since it is already displayed through the commonality of data items. Document Handler is not, however, problem free. It requires a large block of memory to run and as a result is slow in its response. It cannot handle Document Data Item Lists with more than 180 entries. This presents a problem when attempting to use the MERGE and the DOCUMENT SET commands in the process of determining the completeness of the resident documents. During the design of the Ozark Forest database, this problem arose on more than one occasion requiring the designer to decompose the groupings of data into smaller subgroups which could be better handled. In order for this methodology to be effective for future design efforts, a modification to the current version of the program is required which will allow it to handle larger documents and databases.

Since Document Handler produces only the printouts as discussed, the designer is left with the task of producing the data diagrams manually. This proved to be a process which was not as difficult to perform due to the capabilities of the system

to group the Document Entities or records and show the keys for each. Since the data has already been manipulated and contains the minimum of data items and records, the resulting diagrams are less complex and easier to produce than in the first approach. In this project, the diagrams presented no difficulty to produce.

Another problem with Document Handler is the lack of editing capabilities. If the user makes a mistake in entering data, misspells a data item name, omits a data item by mistake or commits any other of many easily made errors, the user must leave Document Handler and utilize a system editor to alleviate the mistakes made during the processing session. To continue processing he must then reenter Document Handler, retrieve the current database file and continue working. This is a time-consuming and frustrating problem which must be resolved if this system is to be an effective design tool. Memory usage might, however, preclude the addition of any large editing capability as the current version already requires more than 150 K-bytes of memory. With a database of any size being processed, the system could easily exceed the current maximum of 500 K-bytes allowed any user on the computer currently being utilized to implement this program.

The first approach would be very difficult to output in printed form. The amount of software required to perform the task of making a readable printout would be prohibitive. Even if it could be programmed, the current capabilities of printers would not allow the printing of large complex diagrams such as the first diagram resulting from this methodology. The system

would have to be able to decompose the larger diagrams into several smaller diagrams while keeping track of the location of each of the small diagrams and which ones were printed and which ones remained to be printed.

In both processes, the determination and isolation of keys required the careful examination of the documents and the document attributes or data items. Often the keys were easily identified but in other instances a new attribute had to be created to uniquely identify the occurrence of a document or record in the database. In some instances data items were concatenated to compose a key. In the first approach, the keys are determined by the designer as the final step. In the second approach keys were identified by the interviewees and the designers during the analysis phase. Both approaches require some knowledge of the target data structure if the keys are to be properly selected to allow the accessing of data from the final structure once it is implemented. Document Handler does not currently allow the designation of candidate keys but rather is restricted to single keys or concatenated keys. This sometimes presents problems in attempting to present functional dependencies which exist but are separate from the dependencies of the single or concatenated key. In Document Handler, keys must be identified early in the process since they are used in the manipulation of the data and in preparing the tentative data base for input into the Bernstein Algorithm. The system does not possess the capability to remove a key once it has been declared without going through the same editing procedure already

discussed. A key may be added without having to leave the program but if it is added to the keys which already exist for a particular document it becomes part of a concatenated key containing the old and new keys.

The methodologies differ also in the creation of records which will eventually become the database. The first approach uses the user documents throughout the process and the resulting model contains those same documents less the deleted ones. The data items contained in these documents are the same ones which were on the documents to begin the process less the deleted ones. In the second approach, the user documents are similarly used to begin the process but as the design procedure progresses the designer has the flexibility to manipulate these documents into fewer documents, rename them as desired and combine documents which serve common functions and where duplication is not needed. Both approaches would reduce the number of documents but the first approach is somewhat constrained in its ability to change the documents until the final steps which allow him to use a MERGE process which is not clearly defined in the master's report. The Document Handler can use its MERGE command which merges two documents into one and removes all duplicated data items. It also has the capability through its ADD and REMOVE commands to add attributes and remove attributes from selected documents. The consideration of combining or merging documents might be important in the final design to preclude unnecessary duplication of data items, records and their associated relationships thereby reducing the size of the database.

Document Handler listings can also be easily modified into a data dictionary although not in the program itself. The listings are organized in such a manner that they can be easily edited by simply adding specifications to the attributes such as field types and sizes, changing the name of the document entity to a record name, removing the separate key entities and adding set or relationship names with the associated information to describe the data structure. Modification of Document Handler to provide a data dictionary generator might, as earlier discussed about the editing capabilities, increase the memory requirements beyond current constraints. It could be modified to organize the data and prepare it for input into a separate data dictionary in much the same manner as the PREPARE command does for the Bernstein Algorithm.

The first approach does not appear to have this capability since its output is the E-R model. The information needed to generate the data dictionary is available in the system as Document Entities, Document Entity Data Item Lists and Relationships. This information could be extracted from the system but listings would have to be formatted and the dictionary prepared once the essential items are removed. This does not appear to be within the scope of the intended system.

Since the second methodology is dependent upon the Bernstein algorithm to produce the final schema, it should be noted here that a significant problem was found in the current version of the Bernstein Algorithm. While not linked to Document Handler, the problem arose when the tentative schema from the Handler was

input into the Bernstein Algorithm Program. It was discovered that the current version cannot handle large databases. It required modification of the program and since the personnel responsible for writing it and those with any experience in using it had either left the university or were unable to be contacted, it required several days and several attempts before the author was able to isolate the changes needed to run larger database input files. Additionally, the output format had to be changed to remove unnecessary items which only served to produce lengthy printouts. Those changes have been documented and are stored in BERN3.P on the computer.

Both of these methodologies require some intuitive knowledge on the part of the designer. The first approach is less dependent than the second since the algorithm constrains the user from having to make many decisions to arrive at the tentative model diagram. The second approach requires the designer to know how the final listing will be modeled and what access strategy will be used to extract data from the final result. Additionally, the second approach allows the designer to manipulate the documents according to his understanding of how the data could be better organized, specifically taking into consideration the usage factors. This flexibility is beneficial if the designer understands the final target structure but can cause problems if he does not. The first approach, while more constrained, produces a model without requiring any knowledge of the target database. This resulting model may not reflect the optimal organization of the data since it is not based upon any

access strategy, usage criteria or the final implementation. The designer must be able to manipulate the final E-R model if these criteria are to be considered.

The use of these two methodologies in future design efforts depends upon several factors. In the case of the first methodology, it must first be implemented if possible. If it can be implemented it would also require modifications such as those already discussed plus enhancements which allow the designer more flexibility. Included among these modifications would be editing capabilities, output capabilities, and interactive diagram manipulation such as reduction techniques. Additionally, the system would have to be modified to handle large data bases which the proposed version cannot do. These modifications are necessary if the designer is to be able to manipulate the diagrams according to his judgment and experience since the system has no basis for making such decisions.

The second methodology must also be modified as already discussed. Additionally, Document Handler would be a more viable design tool if it were able to define the range of attributes, graphically represent the data, automate the database generation process and incorporate the interactive editing capability. The automation of the specification of more than one candidate key, the ability to handle repeating groups plus the capability of merging more than two documents at a time and merging documents by their category (INPUT, OUTPUT, RESIDENT) would further enhance this methodology as an automated design tool.

## CHAPTER 6

### CONCLUSION

This report compares two methodologies previously proposed by two former students to automate the database design process. Both of the methodologies provide aspects which can be integrated into an effective approach which would assist the database designer in the difficult task of taking user documents and manipulating the data contained there into an effective database structure. The design of the Budget and Finance Section of the Ozark Forest database proved to be both a challenging and valuable learning experience. The lessons learned in the use of the two systems will serve as building blocks for future design efforts. The experience gained in designing an actual implementation of a data base will be invaluable should the opportunity to be involved in such a project arise in the future.

This report is not proposed as a model for the use of these systems in future design efforts but rather as an illustration of their use and the lessons learned during this process. These lessons and the recommendations included in the report should serve as a basis for enhancing future design efforts using these systems.

The two systems used require modifications before they can be considered viable design tools. Whether or not the first

methodology can be implemented and the modifications made to both is a matter which will require additional research. This research might well lead to a more effective version of either system or even an integration of the strong aspects of each into an effective automated design tool.

## BIBLIOGRAPHY

- AANS 72      Aanstad, P. S., Skylstad, G., and Solvberg, A. "CASCADE -- A Computer-Based Documentation System," Computer-Aided Information Systems, Analysis and Design, (1972), 93-112.
- AUER 76      Database Design Methodology, Part 1, Auerbach Publishers, Inc., (1976), 1-16.
- BERN 76      Bernstein, Philip A. "Synthesizing Third Normal Form Relations from Functional Dependencies," ACM Transactions on Database Systems, I (December 1976), 277-298.
- CARD 79      Cardenas, Alfonso F., Database Management Systems, Boston, MA., (1979), 407-479.
- CHEN 76      Chen, Peter. "The Entity-Relationship Model: Toward a Unified View of Data," ACM Transactions on Database Systems, I (March 1976), 9-12.
- CODD 78      Codd, E. F. "A Relational Model of Data for Large Shared Data Bases," Communications of the ACM, XIII (June 1978), 377-387.
- COHE 81      Cohen, Meir. "A System For Automatic Generation of Relational Data Bases," Master's Report, Kansas State University, (1981), 1-33.
- FAGI 77      Fagin, R. "Multivalued Dependencies and a New Normal Form for Relational Databases," ACM Transactions on Database Systems, III (Sept 1977), 262-278.
- FINN 77      Finneran, T. R., and Henry, J. S. "Structured Analysis for Data Base Design," Datamation, (Nov 1977), 99-113.
- FISH 81      Fisher, Paul S. "Database Design Technique," Computer Communications, IV (December 1981), 273-280.
- HERS 75      Hershey, E. A., "Problem Statement Language Version 3.0 Language Reference Manual," Working Paper 68, ISDOS Research Project, University of Michigan, (May 1975).

- HUBB 75      Hubbard, G. U. "Technique for Automated Logical Database Design," NYU Symposium on Database Design, (May 1975), 85-90.
- TEOR 82      Teorrey, Toby J., and Fry, James P. Design of Data Base Structures, Englewood Cliffs, CA., 1982, 13-130.
- WOEL 81      Woelk, Darrell W. "The Generation of Entity-Relationship Diagrams from User Documents," Master's Report, Kansas State University, (1981), 1-40.

## APPENDIX A

### SAMPLE DOCUMENT HANDLER LISTING

## DOCUMENTS IN THE SYSTEM

1. BPPD_SF_1143_ADVRTG_ORDER	WITH	3 COLUMNS
2. BPAES_R3_6300_3_BLK_T_PRCH_AR	WITH	6 COLUMNS
3. BPAES_AD_700_REQSTN_FOR_SUP	WITH	18 COLUMNS
4. BPPD_1108_CRTF_IN_LIEU_OF_LS	WITH	15 COLUMNS
5. BPPD_1103_US_GOV_T_BL_OF_LD_G	WITH	15 COLUMNS
6. BPPD_ASCS_441_ORDR_FOR_ARL_P	WITH	8 COLUMNS
7. BPPD_18_RQST_FPR_QTNS	WITH	13 COLUMNS
8. BPPD_AD_633_MLTI_USE_STNDRD	WITH	16 COLUMNS
9. BPPD_AD_838A_PRTL_RCPT_NTFCT	WITH	8 COLUMNS
10. BPPD_AD_838B_INVC_RCPT_CRTF	WITH	17 COLUMNS
11. BPPD_AD_14_RQST_FOR_SUPP_EQP	WITH	10 COLUMNS
12. BPPD_AD_838_9_PRCH_ORDR	WITH	23 COLUMNS

## DOCUMENT / COLUMN LISTING

\* BPPD\_SF\_1143\_ADVRTG\_ORDER

## DOCUMENT ATTRIBUTES :

LOC . OUT

ADV\_ORDER\_NUMBER \*

DATE\_OF\_ORDER

NAME\_OF\_PUB\_ADVERTISED

\* BPAES\_R8\_6300\_3\_BLK\_T\_PRCH\_AR

## DOCUMENT ATTRIBUTES :

LOC . INP

BPA\_NAME \*

BPA\_ADDRESS

ORDER\_NUMBER \*

DATE\_OF\_REPORT

PERSONS\_AUTHORIZED\_TO\_PLACE\_

SCOPE\_OF\_ARRANGEMENT

\* BPAES\_AD\_700\_REQSTN\_FOR\_SUP

## DOCUMENT ATTRIBUTES :

LOC . INP

REQUISITIONING\_OFFICE

RECEIVING\_OFFICE\_NO

CONTRACT\_NUMBER \*

FUND\_CODE

REQUISITION\_NO

REQUISITION\_DATE

VENDOR\_NAME

VENDOR\_ADDRESS

CONSIGNEE\_NAME

CONSIGNEE\_ADDRESS

QUANTITY

LINE\_ITEM

DESCRIPTION

BUDGET\_OBJECT

ACCTNG\_LINE

UNIT\_OF\_ISSUE

UNIT\_PRICE

MANAGEMENT\_CODE\_AMOUNT

\* BPPD\_1108\_CRTF\_IN\_LIEU\_OF\_LS

## DOCUMENT ATTRIBUTES :

LOC . OUT

ORIG\_BILL\_OF\_LADING\_NO \*

TRANSPORTATION\_CO\_TENDERED\_T

FROM\_SHIPPING\_POINT

CONSIGNEE\_NAME

DESTINATION\_NAME

CHARGES\_TO

MANAGEMENT\_CODE

PACKAGES\_NO

PACKAGES\_KIND

DESCRIPTION

NUMBER\_OF\_PACKAGES

WEIGHTS  
CONSIGNEE\_ADDRESS  
DESTINATION\_ADDRESS  
FULL\_NAME\_OF\_SHIPPER

\* BPP0\_1103\_US\_G0VT\_BL\_OF\_LOG

DOCUMENT ATTRIBUTES :

LOC . OUT

TRANSPORTATION\_CO\_TENDERED\_T  
FROM\_SHIPPING\_POINT  
FULL\_NAME\_OF\_SHIPPER \*  
CONSIGNEE\_NAME  
DESTINATION\_NAME  
BILL\_CHARGES\_TO  
MANAGEMENT\_CODE  
PACKAGES\_NO  
PACKAGES\_KIND  
DESCRIPTION  
NUMBER\_OF\_PACKAGES  
WEIGHTS  
CONSIGNEE\_ADDRESS  
BILL\_OF\_LADING\_NO  
DESTINATION\_ADDRESS

\* BPP0\_ASCS\_441\_ORDR\_FOR\_ARL\_P

DOCUMENT ATTRIBUTES :

LOC . OUT

PURCHASE\_ORDER\_NUMBER  
CONSIGNEE\_NAME \*  
CONSIGNEE\_ADDRESS \*  
SIZE\_TYPE\_OF\_REPRODUCTIONS  
QUAN\_EACH  
CODE\_OR\_SYMBOL  
ROLL\_NO  
EXPOSURE\_NO

\* BPP0\_18\_R0ST\_FPR\_QTNS

DOCUMENT ATTRIBUTES :

LOC . OUT

REQUEST\_NO \*  
DATE\_ISSUED  
REQUISITION\_PURCHASE\_NO  
ISSUED\_BY \*  
TO\_NAME \*  
TO\_ADDRESS  
DESTINATION\_NAME  
DESTINATION\_ADDRESS  
ITEM\_NUMBER  
SUPPLIES\_SERVICES  
QUANTITY  
UNIT\_TYPE  
UNIT\_PRICE

\* BPP0\_AD\_633\_MLTI\_USE\_STNDRD\_

DOCUMENT ATTRIBUTES :

LOC . OUT

DOCUMT\_IDENT \*  
ROUTING\_IDENT \*  
M\_AND\_S  
REQUISITIONER  
DATE\_OF\_REPORT  
SIGNAL  
FUND  
SERIAL  
STOCK\_NUMBER  
UNIT\_OF\_ISSUE  
QUANTITY  
UNIT\_PRICE  
MANAGEMENT\_CODE  
OBJ\_CLASS  
ACCTNG\_LINE  
MANAGEMENT\_CODE\_AMOUNT

\* 9PPD\_AD\_838A\_PRTL\_RCPT\_NTFC

DOCUMENT ATTRIBUTES :

LOC . OUT

PURCHASE\_ORDER\_NUMBER \*  
DATE\_OF\_RECEIPT  
RECEIVING\_OFFICE\_NO \*  
MATRL\_OR\_SUPPLIES\_ITEM\_NO  
QUANTITY\_RECEIVED  
UNIT\_OF\_ISSUE  
DESCRIPTION  
RECEIPT\_STATUS

\* 9PPD\_AD\_838B\_INVC\_RCPT\_CRTF

DOCUMENT ATTRIBUTES :

LOC . OUT

PURCHASE\_ORDER\_NUMBER \*  
DATE\_OF\_RECEIPT  
DATE\_INVOICE\_RECEIVED  
VENDOR\_INVOICE\_NUMBER \*  
VENDOR\_NAME  
MATRL\_OR\_SUPPLIES\_ITEM\_NO  
DESCRIPTION  
UNIT\_OF\_ISSUE  
DOLLAR\_AMOUNT  
NON\_MERCHANDISE\_CHARGE  
FREIGHT  
FEDERAL\_EXCISE\_TAX  
STATE\_OR\_LOCAL\_TAX  
TRADE  
DISCOUNT  
CREDIT  
QUANTITY\_RECEIVED

\* 9PPD\_AD\_14\_RQST\_FOR\_SUPP\_EQP

DOCUMENT ATTRIBUTES :

LOC . OUT

DATE\_OF\_REQUEST

MANAGEMENT\_CODE  
ENCUMBERED  
CONSIGNEE\_NAME  
CONSIGNEE\_ADDRESS  
MATRL\_OR\_SUPPLIES\_ITEM\_NO  
DESCRIPTION  
QUANTITY  
UNIT\_OF\_ISSUE      \*  
UNIT\_PRICE

\* BPP0\_AD\_838\_9\_PRCH\_ORDR

DOCUMENT ATTRIBUTES :  
LOC . OUT

ORDER\_DATE  
SF\_37  
FUND\_CODE      \*  
PURCHASE\_ORDER\_NUMBER      \*  
PURCHASE\_ORDER\_SUB  
CONTRACT\_NUMBER  
VENDER\_NAME  
VENDOR\_ADDRESS  
CONSIGNEE\_NAME  
CONSIGNEE\_ADDRESS  
LINE\_ITEM  
ACTION\_CODE  
DESCRIPTION  
BUDGET\_OBJECT  
ACCTNG\_LINE  
QUANTITY  
UNIT\_OF\_ISSUE  
UNIT\_PRICE  
FOB\_POINT  
DISCOUNT\_TERMS  
ESTIMATED\_FREIGHT  
MANAGEMENT\_CODE  
MANAGEMENT\_CODE\_AMOUNT

DOCUMENT HANDLER VER 0.2 RUN

COLUMN CROSS REFERENCE

COLUMNS	IN DOCUMENTS
ADV_ORDER_NUMBER	BPPD_SF_1143_ADVRTG_ORDER
ACCTNG_LINE	BPAES_AD_700_REQSTN_FOR_SUP BPPD_AD_633_MLTI_USE_STNDRD BPPD_AD_838_9_PRCH_ORDR
ACTION_CODE	BPPD_AD_838_9_PRCH_ORDR
BPA_NAME	BPAES_R8_6300_3_BLK_T_PRCH_AR
BPA_ADDRESS	BPAES_R8_6300_3_BLK_T_PRCH_AR
BUDGET_OBJECT	BPAES_AD_700_REQSTN_FOR_SUP BPPD_AD_838_9_PRCH_ORDR
BILL_CHARGES_TO	BPPD_1103_US_GOVT_BL_OF_LDG
BILL_OF_LADING_NO	BPPD_1103_US_GOVT_BL_OF_LDG
CONTRACT_NUMBER	BPAES_AD_700_REQSTN_FOR_SUP BPPD_AD_838_9_PRCH_ORDR
CONSIGNEE_NAME	BPAES_AD_700_REQSTN_FOR_SUP BPPD_1108_CRTF_IN_LIEU_OF_LS BPPD_1103_US_GOVT_BL_OF_LDG BPPD_ASCS_441_ORDR_FOR_ARL_P BPPD_AD_14_RQST_FOR_SUPP_EQP BPPD_AD_838_9_PRCH_ORDR
CONSIGNEE_ADDRESS	BPAES_AD_700_REQSTN_FOR_SUP BPPD_1108_CRTF_IN_LIEU_OF_LS BPPD_1103_US_GOVT_BL_OF_LDG BPPD_ASCS_441_ORDR_FOR_ARL_P BPPD_AD_14_RQST_FOR_SUPP_EQP BPPD_AD_838_9_PRCH_ORDR
CHARGES_TO	BPPD_1108_CRTF_IN_LIEU_OF_LS
CODE_OR_SYMBOL	BPPD_ASCS_441_ORDR_FOR_ARL_P
CREDIT	BPPD_AD_8388_INVC_RCPT_CRTF
DATE_OF_ORDER	BPPD_SF_1143_ADVRTG_ORDER
DATE_OF_REPORT	BPAES_R8_6300_3_BLK_T_PRCH_AR BPPD_AD_633_MLTI_USE_STNDRD

DOCUMENT	HANDLER	VER 0.2	RUN
DESCRIPTION	BPAES_AD_700_REQSTN_FOR_SUP BPPD_1108_CRTE_IN_LIEU_OF_LS BPPD_1103_US_GOVTL_BL_OF_LDG BPPD_AD_838A_PRTL_RCPT_NTFCT BPPD_AD_838B_INVC_RCPT_CRTE BPPD_AD_14_RQST_FOR_SUPP_EQP BPPD_AD_838_9_PRCH_ORDR		
DESTINATION_NAME	BPPD_1108_CRTE_IN_LIEU_OF_LS BPPD_1103_US_GOVTL_BL_OF_LDG BPPD_18_RQST_FPR_QTNS		
DESTINATION_ADDRESS	BPPD_1108_CRTE_IN_LIEU_OF_LS BPPD_1103_US_GOVTL_BL_OF_LDG BPPD_18_RQST_FPR_QTNS		
DATE_ISSUED	BPPD_18_RQST_FPR_QTNS		
DOCUMT_IDENT	BPPD_AD_633_MLTI_USE_STNDRD_		
DATE_OF_RECEIPT	BPPD_AD_838A_PRTL_RCPT_NTFCT BPPD_AD_838B_INVC_RCPT_CRTE		
DATE_INVOICE_RECEIVED	BPPD_AD_838B_INVC_RCPT_CRTE		
DOLLAR_AMOUNT	BPPD_AD_838B_INVC_RCPT_CRTE		
DISCOUNT	BPPD_AD_838B_INVC_RCPT_CRTE		
DATE_OF_REQUEST	BPPD_AD_14_RQST_FOR_SUPP_EQP		
DISCOUNT_TERMS	BPPD_AD_838_9_PRCH_ORDR		
EXPOSURE_NO	BPPD_ASCS_441_ORDR_FOR_ARL_P		
ENCUMBERED	BPPD_AD_14_RQST_FOR_SUPP_EQP		
ESTIMATED_FREIGHT	BPPD_AD_838_9_PRCH_ORDR		
FUND_CODE	BPAES_AD_700_REQSTN_FOR_SUP BPPD_AD_838_9_PRCH_ORDR		
FROM_SHIPPING_POINT	BPPD_1108_CRTE_IN_LIEU_OF_LS BPPD_1103_US_GOVTL_BL_OF_LDG		
FULL_NAME_OF_SHIPPER	BPPD_1108_CRTE_IN_LIEU_OF_LS BPPD_1103_US_GOVTL_BL_OF_LDG		
FUND	BPPD_AD_633_MLTI_USE_STNDRD_		
FREIGHT	BPPD_AD_838B_INVC_RCPT_CRTE		
FEDERAL_EXCISE_TAX	BPPD_AD_838B_INVC_RCPT_CRTE		
FOB_POINT	BPPD_AD_838_9_PRCH_ORDR		

ISSUED_BY	BPPD_18_RQST_FPR_QTNS
ITEM_NUMBER	BPPD_18_RQST_FPR_QTNS
LINE_ITEM	BPAES_AD_700_REQSTN_FOR_SUP BPPD_AD_838_9_PRCH_ORDR
MANAGEMENT_CODE_AMOUNT	BPAES_AD_700_REQSTN_FOR_SUP BPPD_AD_633_MLTI_USE_STNDRD BPPD_AD_838_9_PRCH_ORDR
MANAGEMENT_CODE	BPPD_1108_CRTF_IN_LIEU_OF_LS BPPD_1103_US_GOVT_BL_OF_LDG BPPD_AD_633_MLTI_USE_STNDRD BPPD_AD_14_RQST_FOR_SUPP_EQP BPPD_AD_838_9_PRCH_ORDR
M_AND_S	BPPD_AD_633_MLTI_USE_STNDRD
MATRL_OR_SUPPLIES_ITEM_NO	BPPD_AD_838A_PRTL_RCPT_NTFCT BPPD_AD_838B_INVC_RCPT_CRTF BPPD_AD_14_RQST_FOR_SUPP_EQP
NAME_OF_PUB_ADVERTISED	BPPD_SF_1143_ADVRTG_ORDER
NUMBER_OF_PACKAGES	BPPD_1108_CRTF_IN_LIEU_OF_LS BPPD_1103_US_GOVT_BL_OF_LDG
NON_MERCHANDISE_CHARGE	BPPD_AD_838B_INVC_RCPT_CRTF
ORDER_NUMBER	BPAES_R8_6300_3_BLK_T_PRCH_AR
ORIG_BILL_OF_LADING_NO	BPPD_1108_CRTF_IN_LIEU_OF_LS
OBJ_CLASS	BPPD_AD_633_MLTI_USE_STNDRD
ORDER_DATE	BPPD_AD_838_9_PRCH_ORDR
PERSONS_AUTHORIZED_TO_PLACE_	BPAES_R8_6300_3_BLK_T_PRCH_AR
PACKAGES_NO	BPPD_1108_CRTF_IN_LIEU_OF_LS BPPD_1103_US_GOVT_BL_OF_LDG
PACKAGES_KIND	BPPD_1108_CRTF_IN_LIEU_OF_LS BPPD_1103_US_GOVT_BL_OF_LDG
PURCHASE_ORDER_NUMBER	BPPD_ASCS_441_ORDR_FOR_ARL_P BPPD_AD_838A_PRTL_RCPT_NTFCT BPPD_AD_838B_INVC_RCPT_CRTF BPPD_AD_838_9_PRCH_ORDR
PURCHASE_ORDER_SUB	BPPD_AD_838_9_PRCH_ORDR
QUANTITY	BPAES_AD_700_REQSTN_FOR_SUP

DOCUMENT	HANDLER	VER 0.2	RUN
			BPPD_18_RQST_FPR_QTNS BPPD_AD_633_MLTI_USE_STNDRD_ BPPD_AD_14_RQST_FOR_SUPP_EQP BPPD_AD_838_9_PRCH_ORDR
QUAN_EACH			BPPD_ASCS_441_ORDR_FOR_ARL_P
QUANTITY_RECEIVED			BPPD_AD_838A_PRTL_RCPT_NTFCT BPPD_AD_838B_INVC_RCPT_CRTF
REQUISITIONING_OFFICE			BPAES_AD_700_REQSTN_FOR_SUP
RECEIVING_OFFICE_NO			BPAES_AD_700_REQSTN_FOR_SUP BPPD_AD_838A_PRTL_RCPT_NTFCT
REQUISITION_NO			BPAES_AD_700_REQSTN_FOR_SUP
REQUISITION_DATE			BPAES_AD_700_REQSTN_FOR_SUP
ROLL_NO			BPPD_ASCS_441_ORDR_FOR_ARL_P
REQUEST_NO			BPPD_18_RQST_FPR_QTNS
REQUISITION_PURCHASE_NO			BPPD_18_RQST_FPR_QTNS
ROUTING_IDENT			BPPD_AD_633_MLTI_USE_STNDRD_
REQUISITIONER			BPPD_AD_633_MLTI_USE_STNDRD_
RECEIPT_STATUS			BPPD_AD_833A_PRTL_RCPT_NTFCT
SCOPE_OF_ARRANGEMENT			BPAES_R8_6300_3_BLK_T_PRCH_AR
SIZE_TYPE_OF_REPRODUCTIONS			BPPD_ASCS_441_ORDR_FOR_ARL_P
SUPPLIES_SERVICES			BPPD_18_RQST_FPR_QTNS
SIGNAL			BPPD_AD_633_MLTI_USE_STNDRD_
SERIAL			BPPD_AD_633_MLTI_USE_STNDRD_
STOCK_NUMBER			BPPD_AD_633_MLTI_USE_STNDRD_
STATE_OR_LOCAL_TAX			BPPD_AD_838B_INVC_RCPT_CRTF
SF_37			BPPD_AD_838_9_PRCH_ORDR
TRANSPORTATION_CO_TENDERED_T			BPPD_1108_CRTF_IN_LIEU_OF_LS BPPD_1103_US_GOV_T_BL_OF_LDG
TO_NAME			BPPD_18_RQST_FPR_QTNS
TO_ADDRESS			BPPD_18_RQST_FPR_QTNS
TRADE			BPPD_AD_838B_INVC_RCPT_CRTF

UNIT\_OF\_ISSUE

BPAES\_AD\_700\_REQSTN\_FOR\_SUP  
BPPD\_AD\_633\_MLTI\_USE\_STNDRD  
BPPD\_AD\_838A\_PRTL\_RCPT\_NTFT  
BPPD\_AD\_838B\_INVC\_RCPT\_CRTF  
BPPD\_AD\_14\_REQST\_FOR\_SUPP\_EQP  
BPPD\_AD\_838\_9\_PRCH\_ORDR

UNIT\_PRICE

BPAES\_AD\_700\_REQSTN\_FOR\_SUP  
BPPD\_18\_REQST\_FPR\_QTNS  
BPPD\_AD\_633\_MLTI\_USE\_STNDRD  
BPPD\_AD\_14\_REQST\_FOR\_SUPP\_EQP  
BPPD\_AD\_838\_9\_PRCH\_ORDR

UNIT\_TYPE

BPPD\_18\_REQST\_FPR\_QTNS

VENDOR\_NAME

BPAES\_AD\_700\_REQSTN\_FOR\_SUP  
BPPD\_AD\_838B\_INVC\_RCPT\_CRTF

VENDOR\_ADDRESS

BPAES\_AD\_700\_REQSTN\_FOR\_SUP  
BPPD\_AD\_838\_9\_PRCH\_ORDR

VENDOR\_INVOICE\_NUMBER

BPPD\_AD\_838B\_INVC\_RCPT\_CRTF

VENDER NAME

BPPD\_AD\_838\_9\_PRCH\_ORDR

WEIGHTS

BPPD\_1103\_CRTF\_IN\_LIEU\_OF\_LS  
BPPD\_1103\_US\_GOVT\_BL\_OF\_LDG

## APPENDIX B

E-R MODEL GENERATION DIAGRAMS

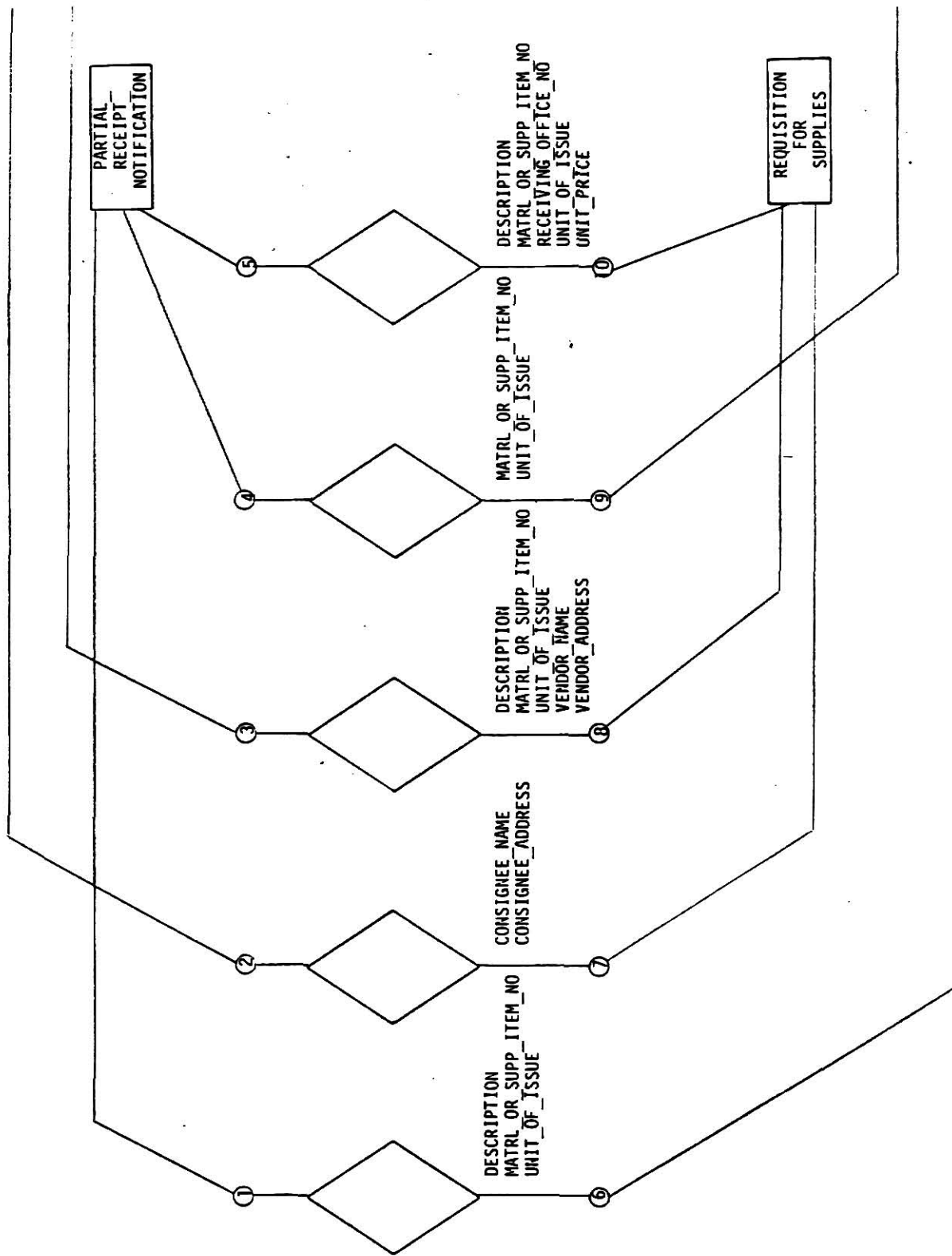


Figure 1



Figure 1a

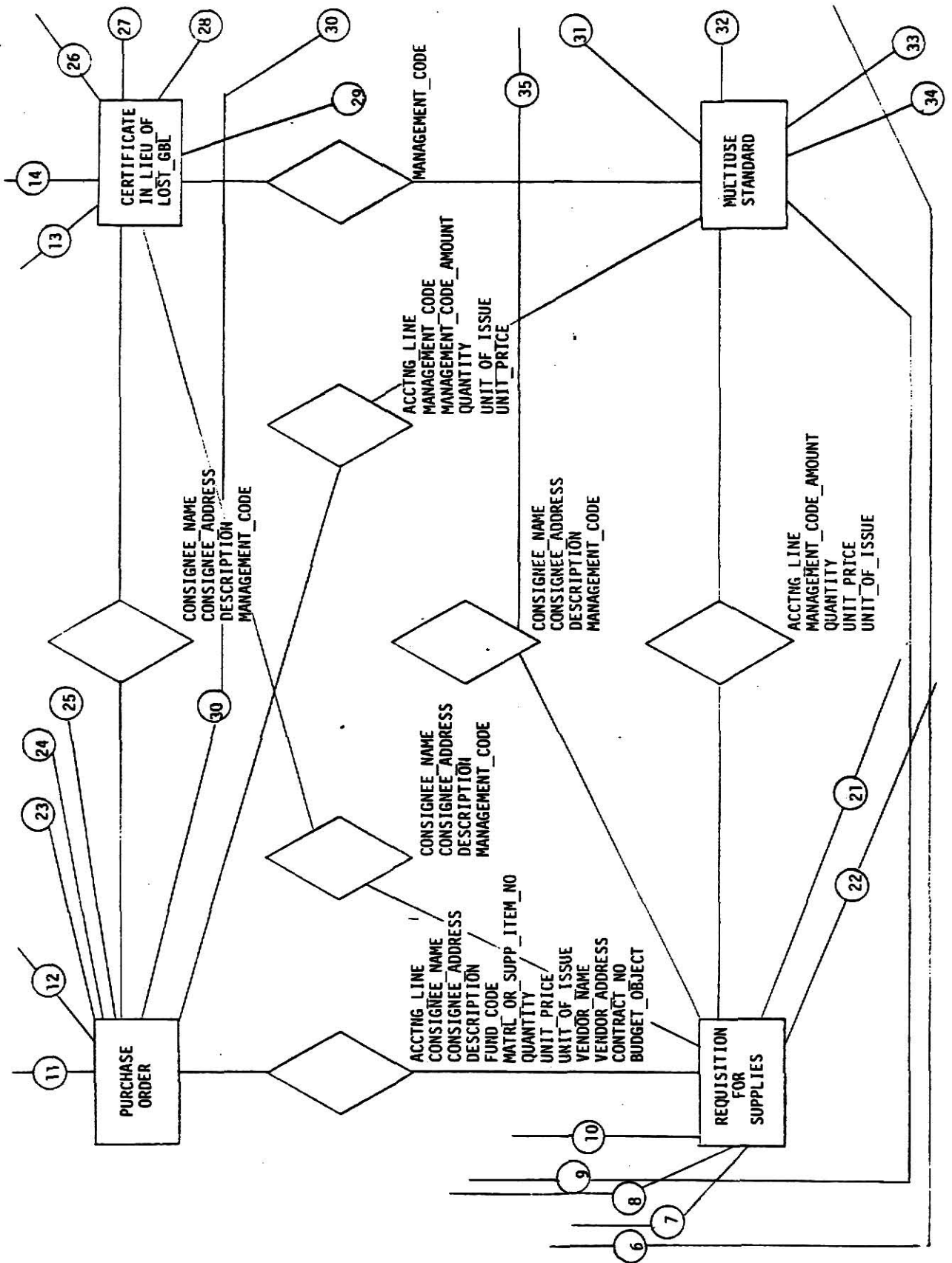
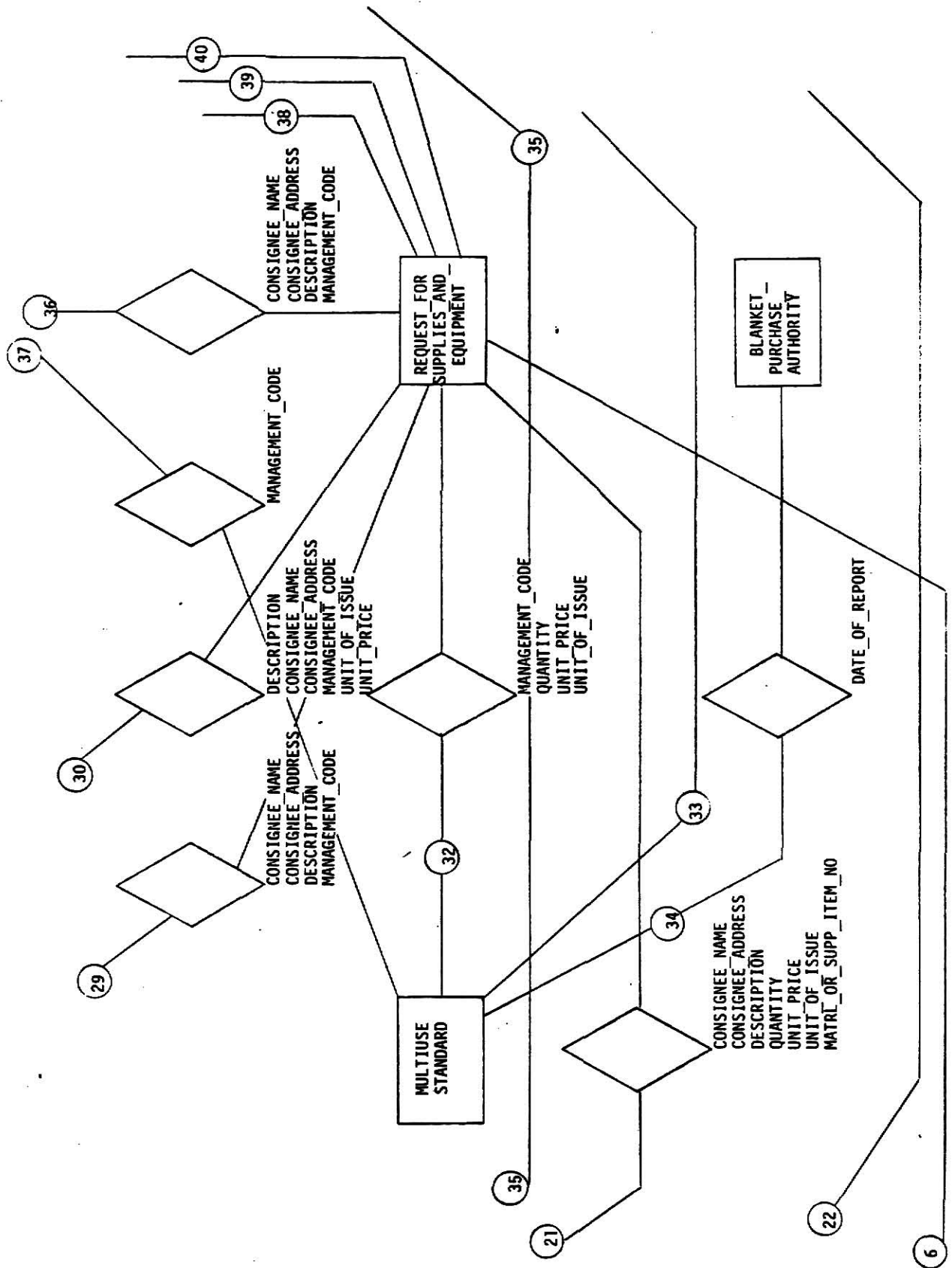


Figure 1b





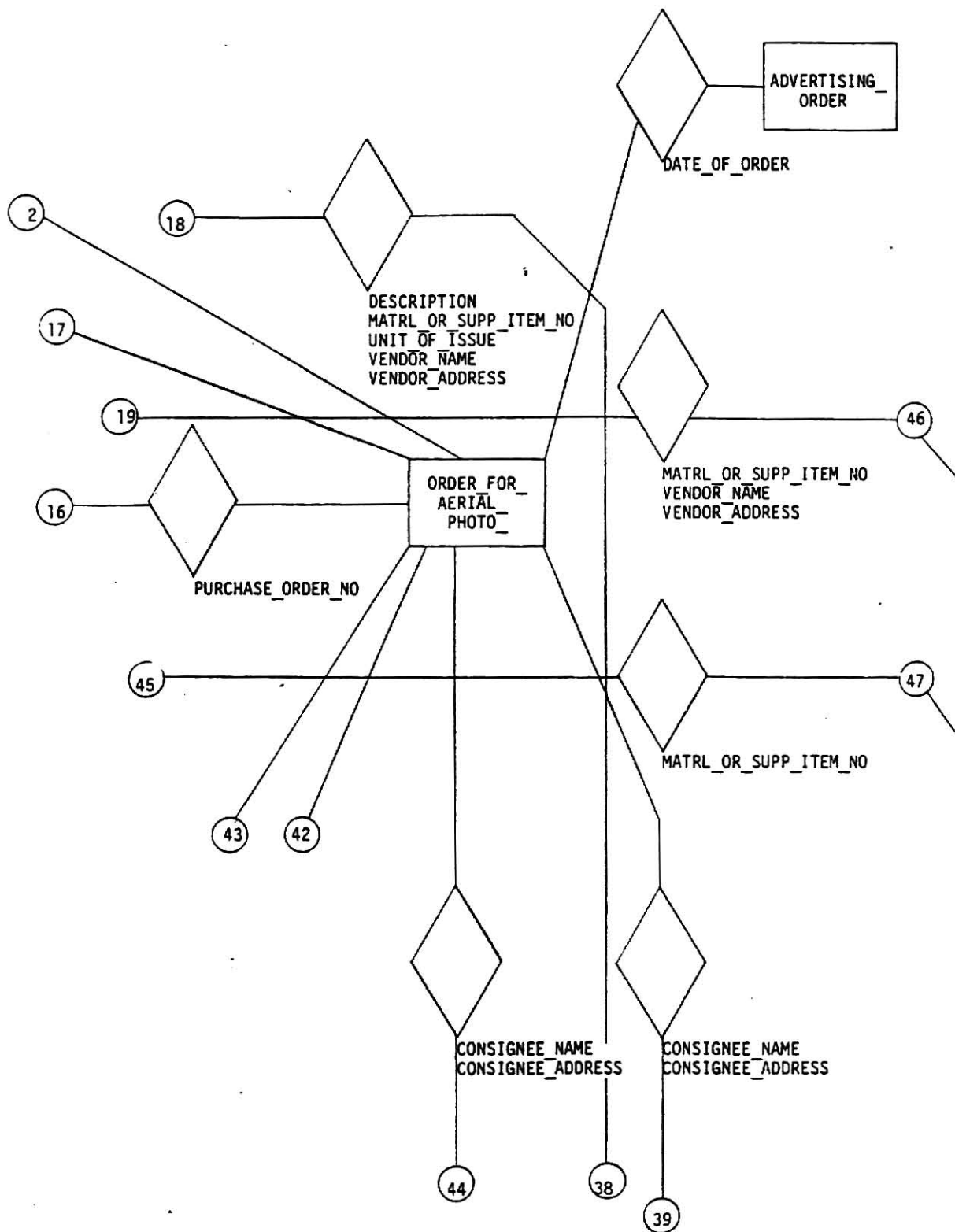


Figure 1e

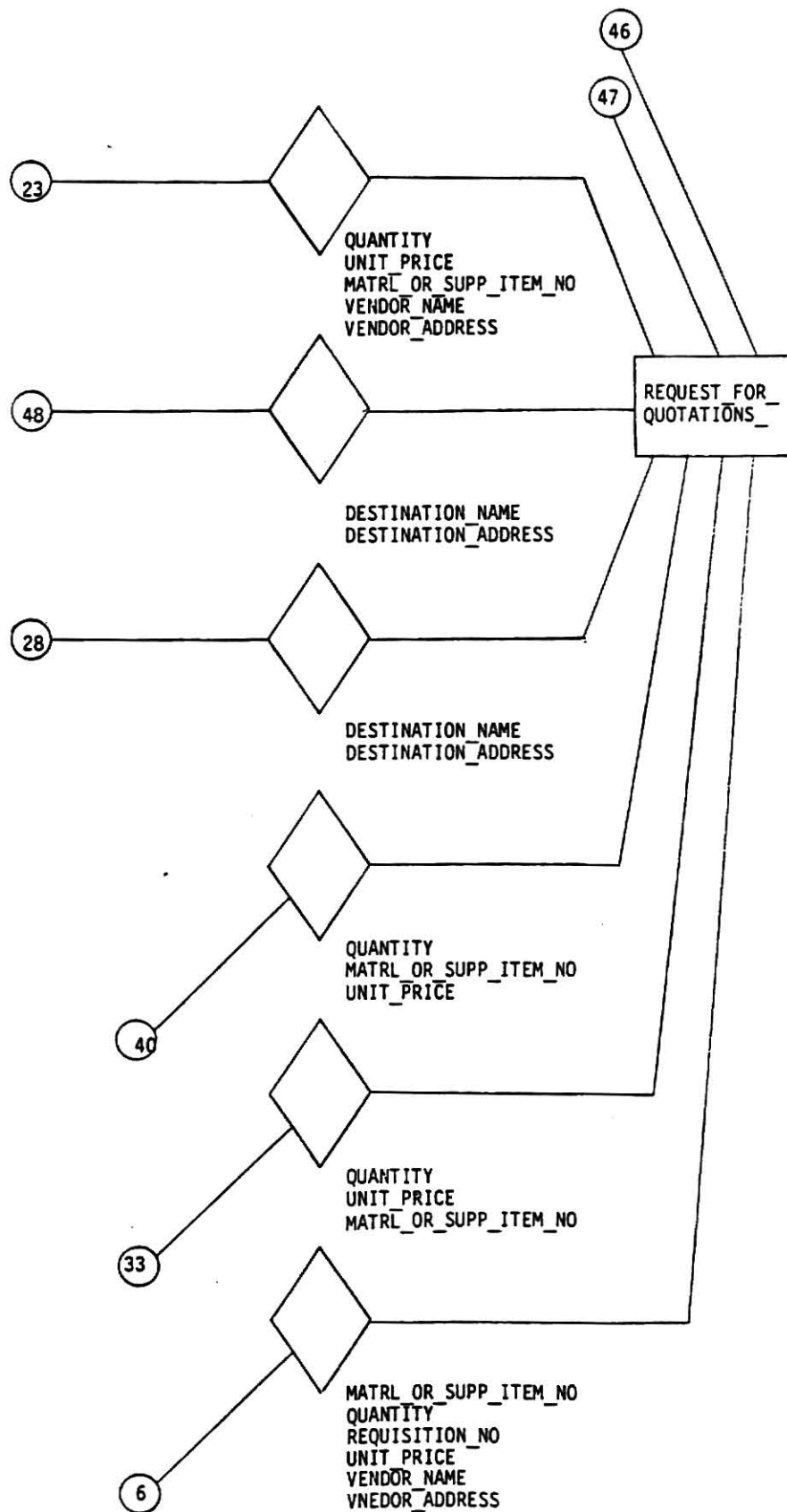


Figure 1f

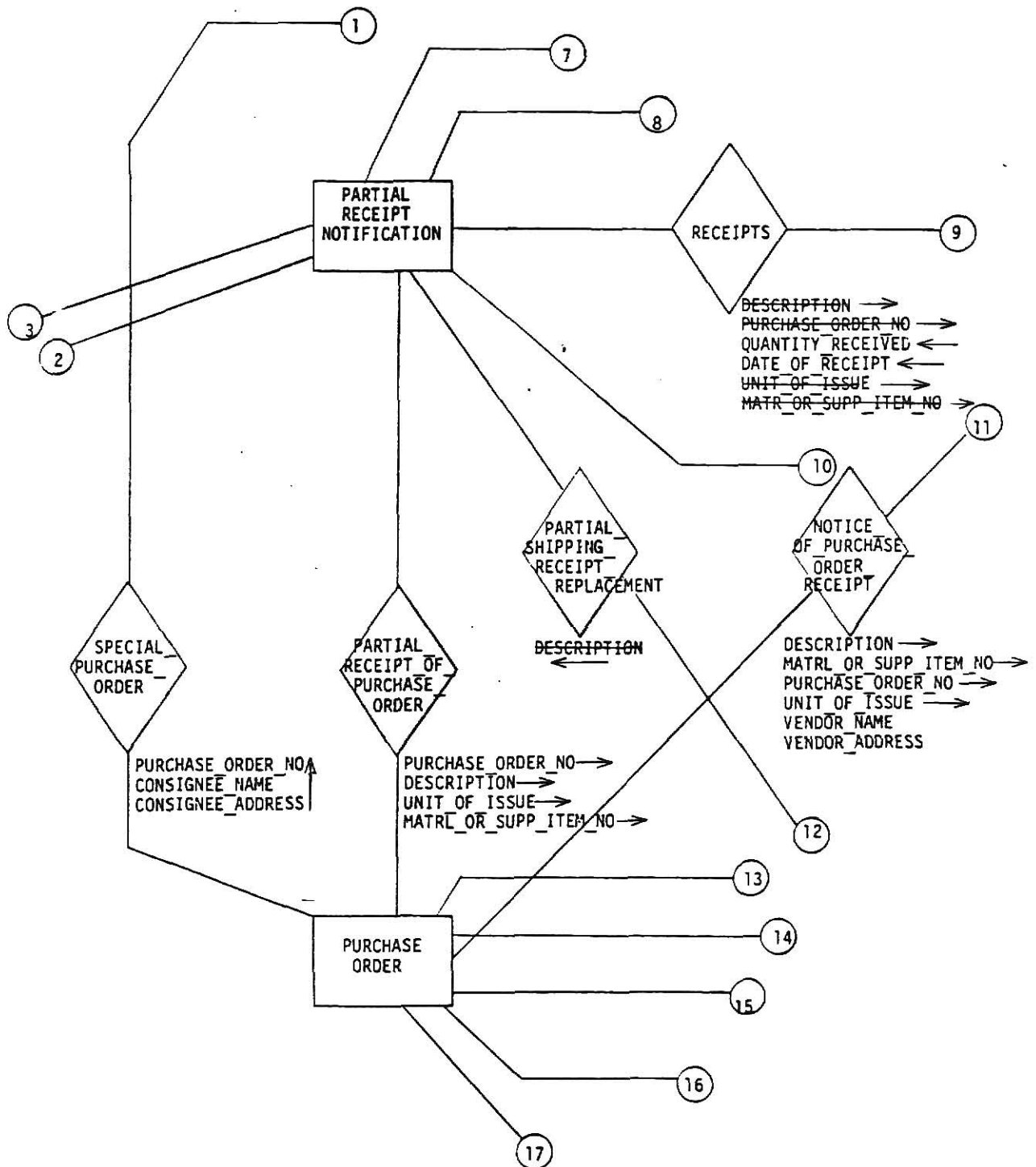


Figure 2

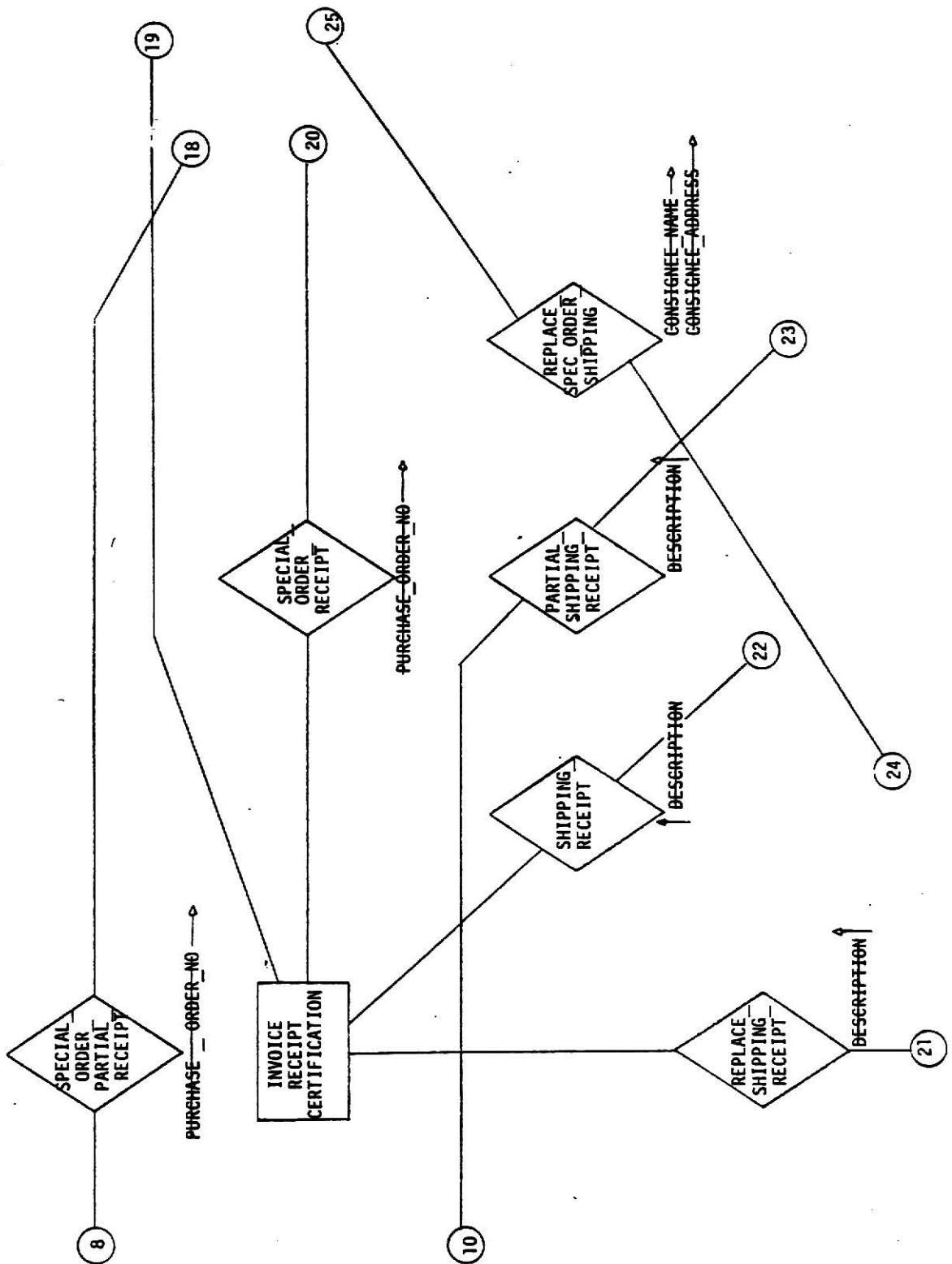


Figure 2a

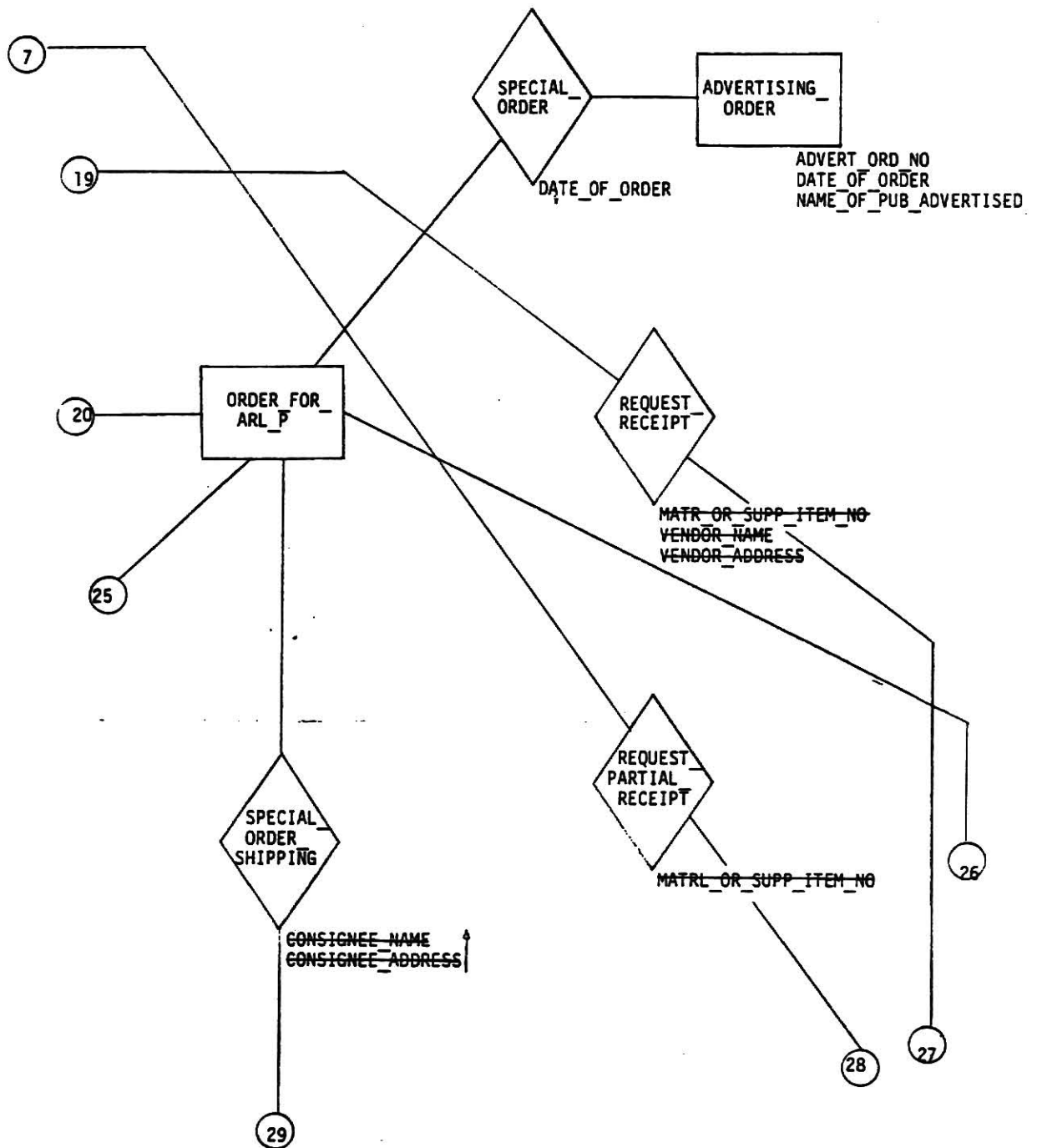


Figure 2b

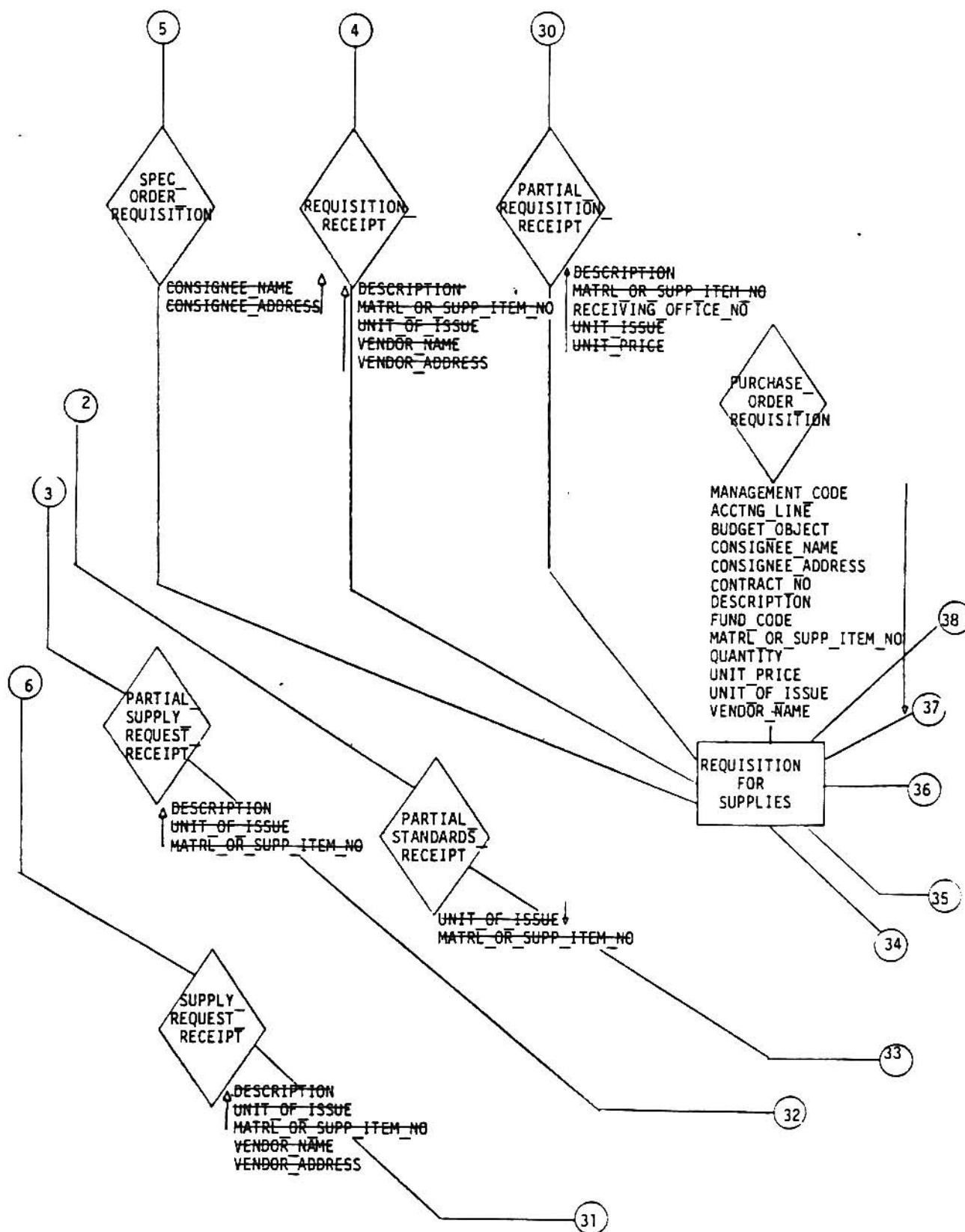


Figure 2c

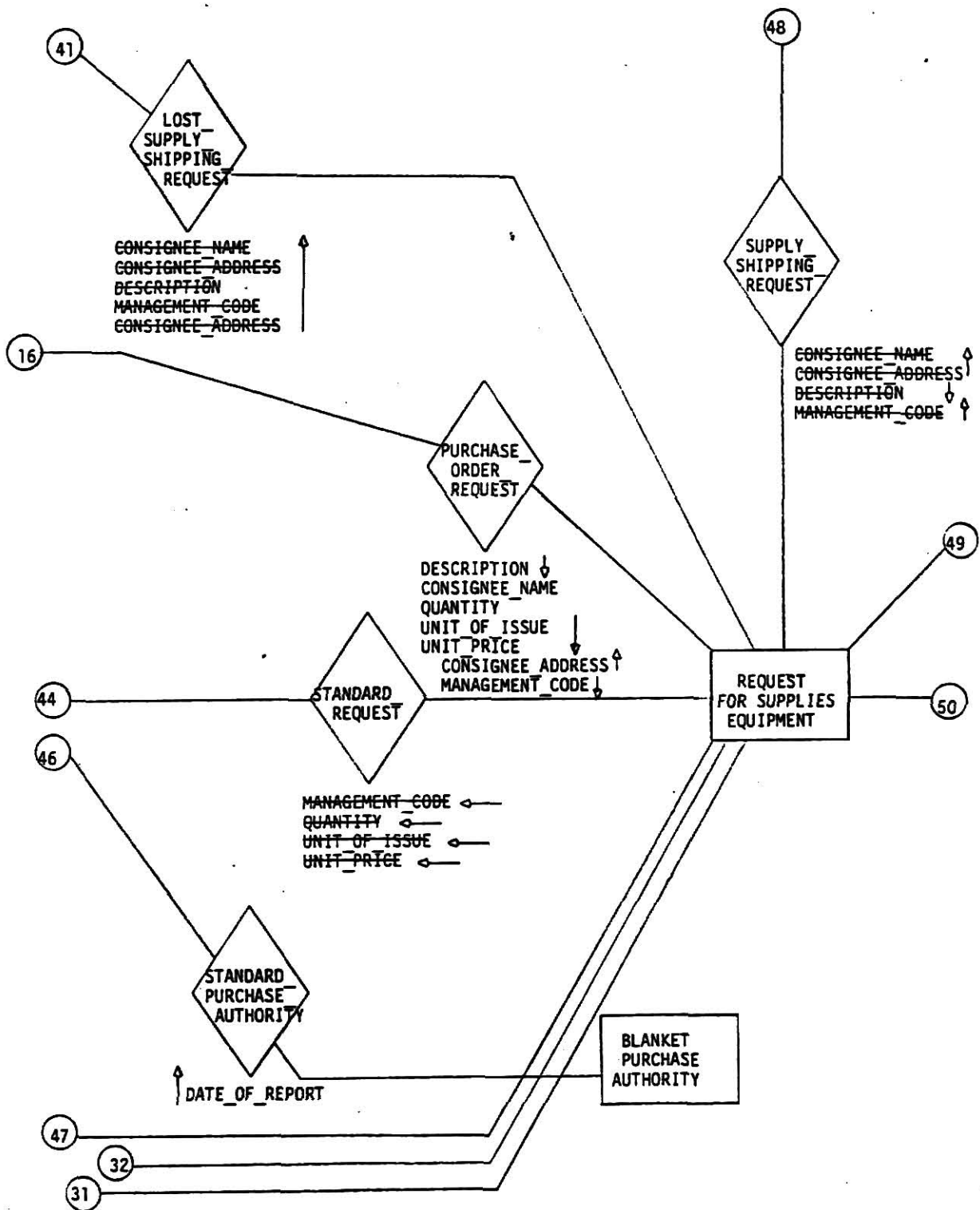


Figure 2d

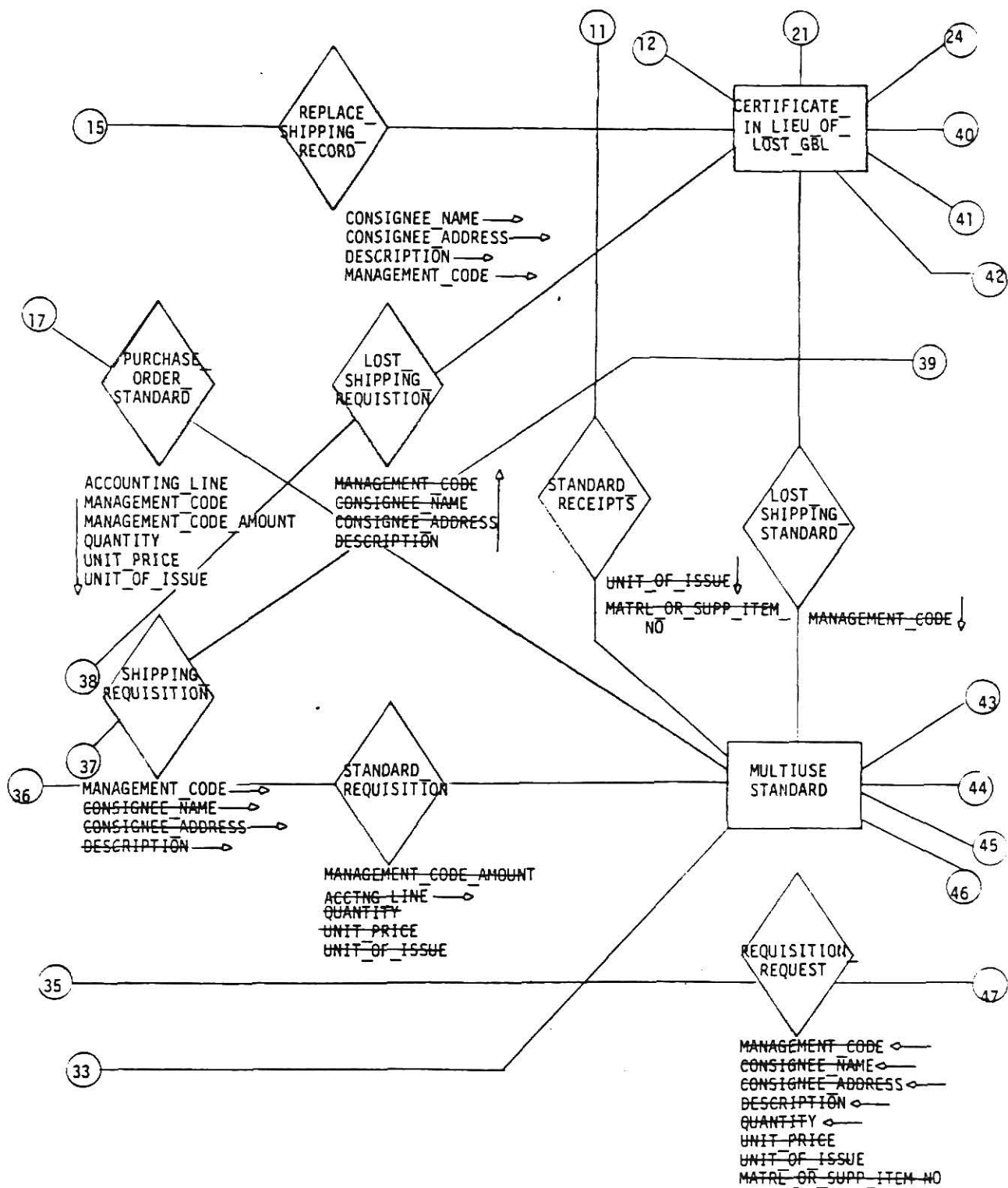


Figure 2e

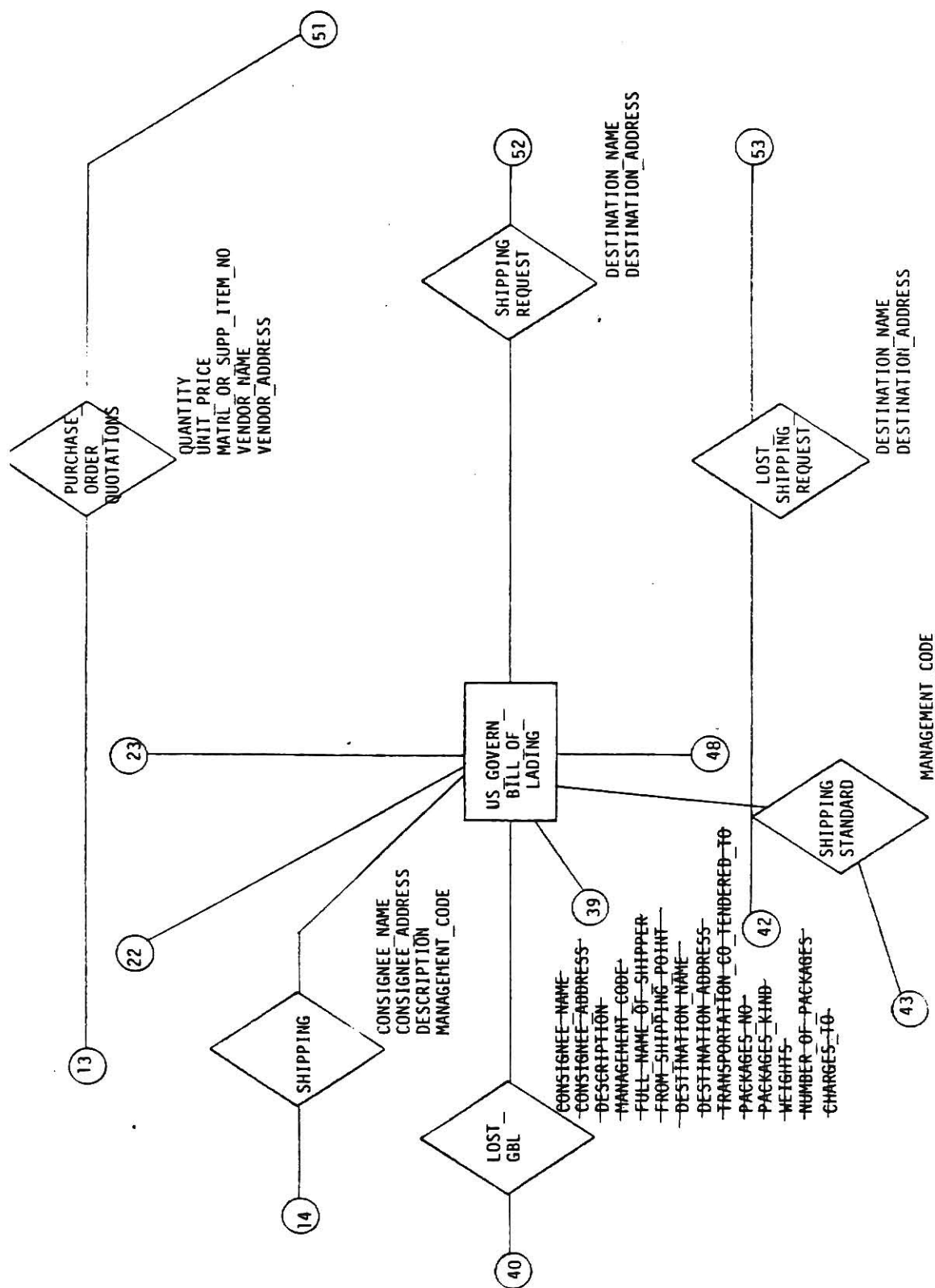


Figure 2f



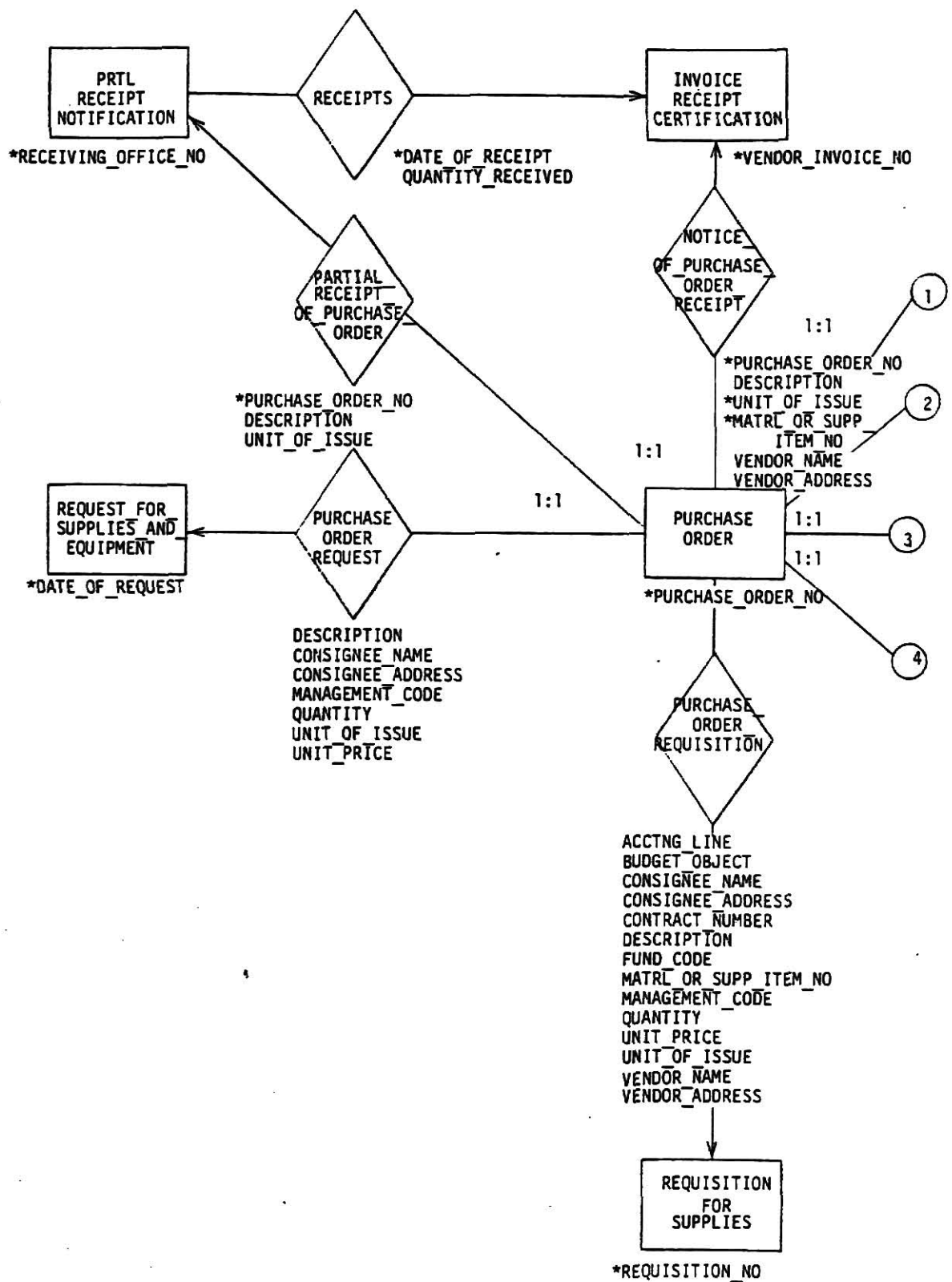


Figure 3

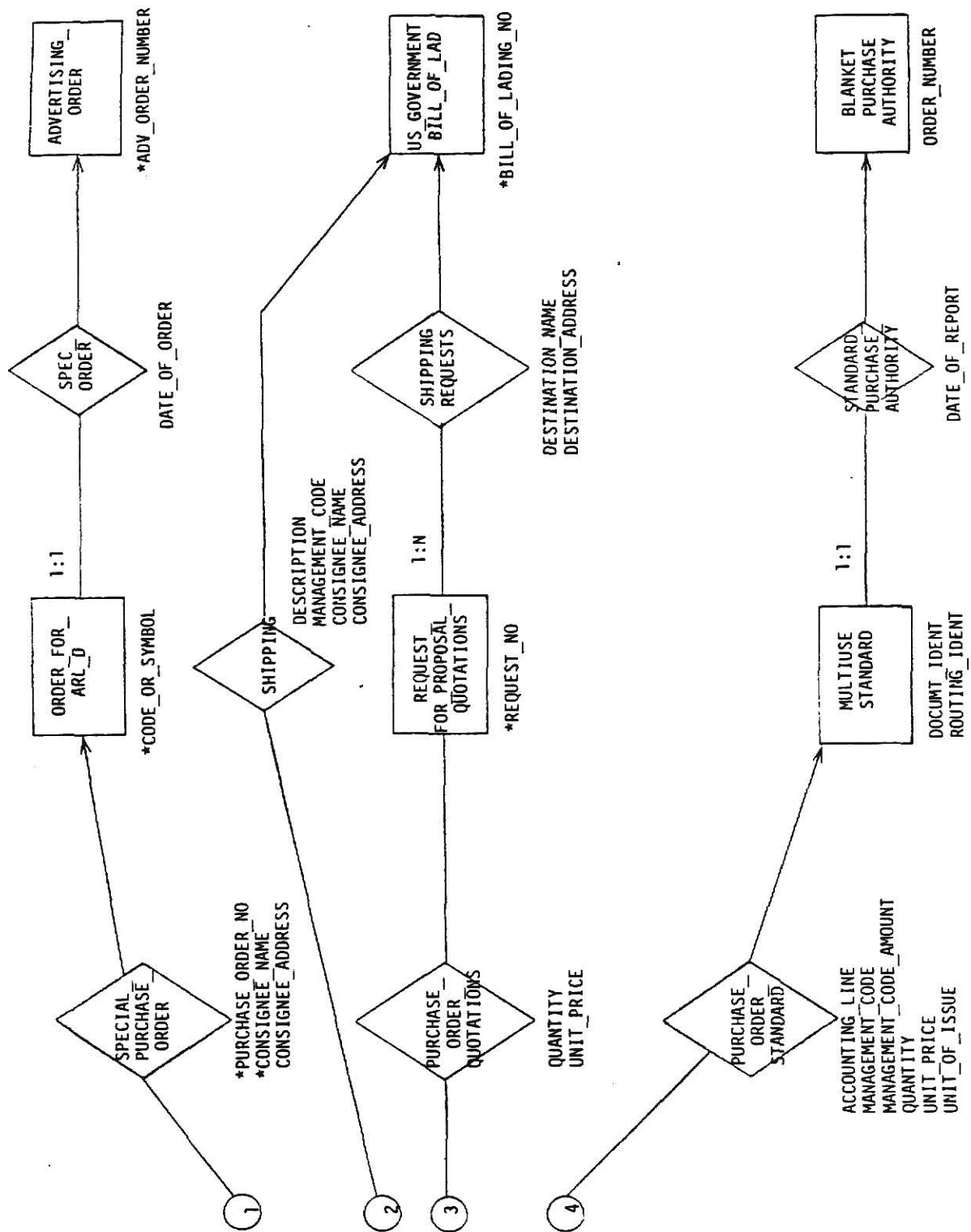


Figure 3a

## APPENDIX C

### DOCUMENT ENTITY WORKING LIST

## DOCUMENT / COLUMN LISTING

\* BPP0\_SF\_1143\_ADVRTG\_ORDER

## DOCUMENT ATTRIBUTES :

LOC . OUT

ADV\_ORDER\_NUMBER \*

DATE\_OF\_ORDER

NAME\_OF\_PUB\_ADVERTISED

\* BPAES\_R8\_6300\_3\_BLK\_T\_PRCH\_AR

## DOCUMENT ATTRIBUTES :

LOC . INP

BPA\_NAME \*

BPA\_ADDRESS

ORDER\_NUMBER \*

~~DATE\_OF\_REPORT~~

PERSONS\_AUTHORIZED\_TO\_PLACE\_

SCOPE\_OF\_ARRANGEMENT

\* BPAES\_AD\_700\_REQSTN\_FGR\_SUP

## DOCUMENT ATTRIBUTES :

LOC . INP

REQUISITIONING\_OFFICE

RECEIVING\_OFFICE\_NO

~~CONTRACT\_NUMBER~~ \*

FUND\_CODE

REQUISITION\_NO

REQUISITION\_DATE

~~VENDOR\_NAME~~~~VENDOR\_ADDRESS~~

CONSIGNEE\_NAME

CONSIGNEE\_ADDRESS

~~QUANTITY~~

LINE\_ITEM

~~DESCRIPTION~~~~BUDGET\_OBJECT~~~~ACCTNG\_LINE~~~~UNIT\_OF\_ISSUE~~~~UNIT\_PRICE~~

MANAGEMENT\_CODE\_AMOUNT

~~\* BPP0\_1100\_GRTF\_IN\_LIEU\_OF\_LS~~

## DOCUMENT ATTRIBUTES :

LOC . OUT

~~ORIG\_BILL\_OF\_LADING\_NO~~ \*~~TRANSPORTATION\_CC\_TENDERED\_T~~~~FROM SHIPPING POINT~~~~CONSIGNEE\_NAME~~~~DESTINATION NAME~~~~CHARGES TO~~~~MANAGEMENT\_CODE~~~~PACKAGES\_NO~~~~PACKAGES\_KIND~~~~DESCRIPTION~~~~NUMBER\_OF\_PACKAGES~~

~~WEIGHTS~~  
~~CONSIGNEE\_ADDRESS~~  
~~DESTINATION\_ADDRESS~~  
~~FULL\_NAME\_OF\_SHIPPER~~

\* BPPD\_1103\_US\_GGVT\_BL\_OF\_LOG

DOCUMENT ATTRIBUTES :

LOC . OUT

TRANSPORTATION\_CO\_TENDERED\_T  
FROM\_SHIPPING\_POINT  
FULL\_NAME\_OF\_SHIPPER \*

~~CONSIGNEE\_NAME~~  
DESTINATION\_NAME  
BILL\_CHARGES\_TO  
~~MANAGEMENT\_CODE~~  
PACKAGES\_NO  
PACKAGES\_KIND  
~~DESCRIPTION~~  
NUMBER\_OF\_PACKAGES  
WEIGHTS  
~~CONSIGNEE\_ADDRESS~~  
BILL\_OF\_LADING\_NO  
DESTINATION\_ADDRESS

\* BPPD\_ASCS\_441\_ORDR\_FOR\_ARL\_P

DOCUMENT ATTRIBUTES :

LOC . OUT

~~PURCHASE\_ORDER\_NUMBER~~  
~~CONSIGNEE\_NAME~~ \*  
~~CONSIGNEE\_ADDRESS~~ \*  
SIZE\_TYPE\_OF\_REPRODUCTIONS  
QUAN\_EACH  
CODE\_OR\_SYMBOL  
ROLL\_NO  
EXPOSURE\_NO

\* BPPD\_18\_RQST\_FPR\_QTNS

DOCUMENT ATTRIBUTES :

LOC . OUT

REQUEST\_NO \*

DATE\_ISSUED  
REQUISITION\_PURCHASE\_NO  
ISSUED\_BY \*

~~TO\_NAME~~ \*  
~~TO\_ADDRESS~~  
~~DESTINATION\_NAME~~  
~~DESTINATION\_ADDRESS~~  
~~ITEM\_NUMBER~~  
SUPPLIES\_SERVICES  
QUANTITY  
UNIT\_TYPE  
~~UNIT\_PRICE~~

\* BPPD\_AD\_633\_MLTI\_USE\_STNDRD\_

DOCUMENT ATTRIBUTES :

~~MANAGEMENT\_CODE~~  
~~ENCUMBERED~~  
~~CONSIGNEE\_NAME~~  
~~CONSIGNEE\_ADDRESS~~  
~~MATRL\_OR\_SUPPLIES\_ITEM\_NO~~  
~~DESCRIPTION~~  
~~QUANTITY~~  
~~UNIT\_OF\_ISSUE~~ \*  
~~UNIT\_PRICE~~

\* BPPD\_AD\_838\_9\_PRCH\_ORDR

DOCUMENT ATTRIBUTES :  
LOC . OUT

ORDER\_DATE  
SF\_37  
FUND\_CODE \*  
PURCHASE\_ORDER\_NUMBER \*  
PURCHASE\_ORDER\_SUB  
CONTRACT\_NUMBER  
VENDER\_NAME  
VENDOR\_ADDRESS  
CONSIGNEE\_NAME  
CONSIGNEE\_ADDRESS  
LINE\_ITEM  
ACTION\_CODE  
DESCRIPTION  
BUDGET\_OBJECT  
ACCTNG\_LINE  
QUANTITY  
UNIT\_OF\_ISSUE  
UNIT\_PRICE  
FOB\_POINT  
DISCOUNT\_TERMS  
ESTIMATED\_FREIGHT  
MANAGEMENT\_CODE  
MANAGEMENT\_CODE\_AMOUNT

## **APPENDIX D**

### **BUDGET / FINANCE DATA STRUCTURE DIAGRAM**

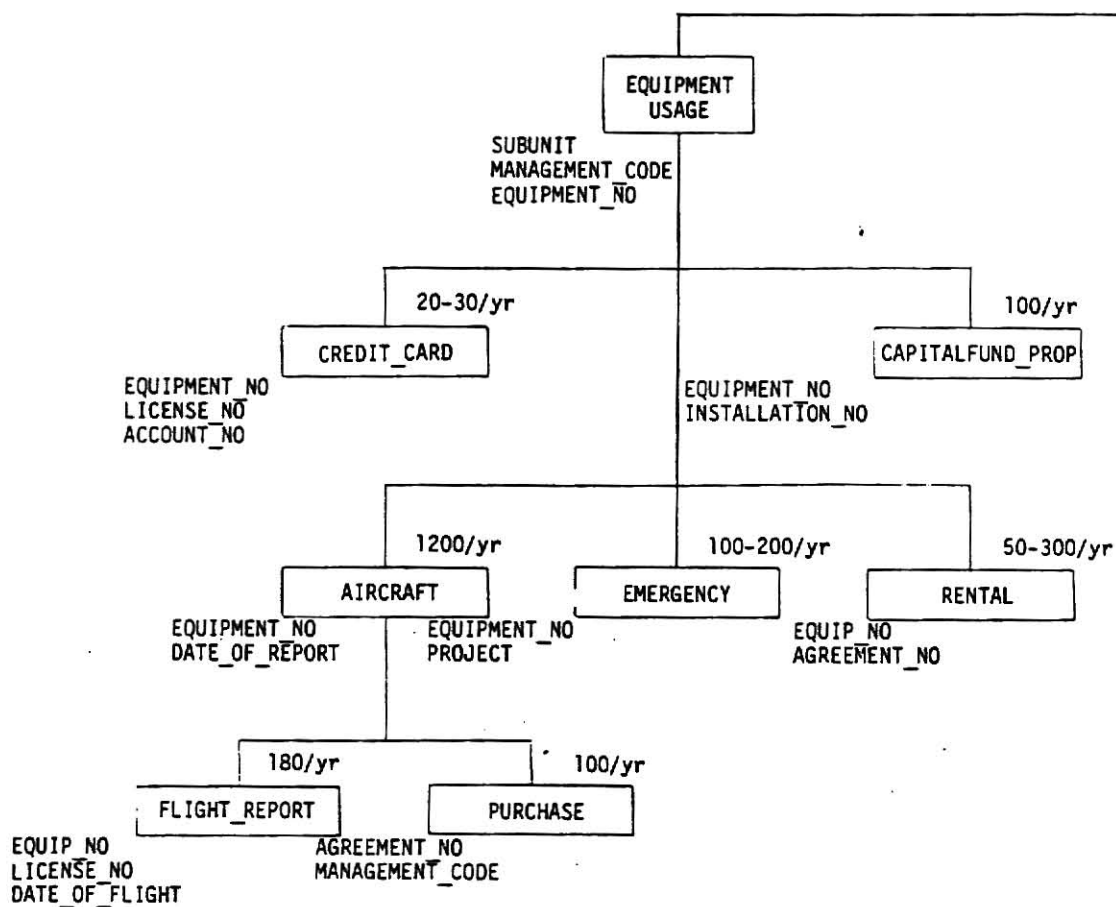


Figure 1  
EQUIPMENT USAGE

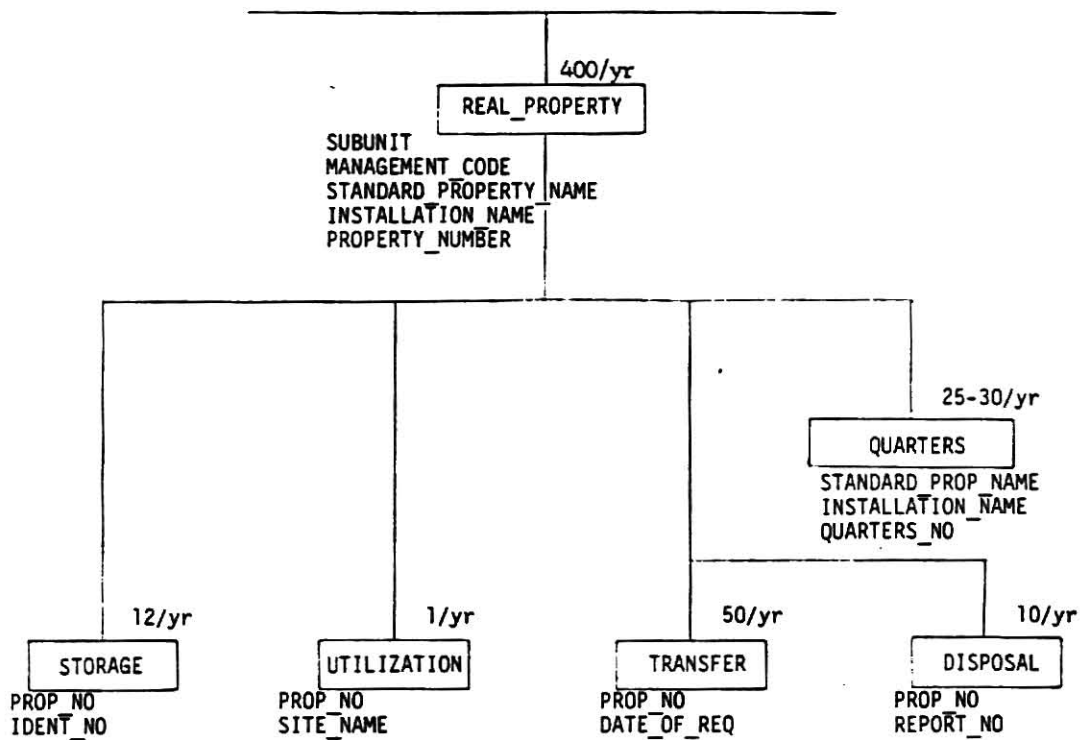


Figure 2  
REAL\_PROPERTY

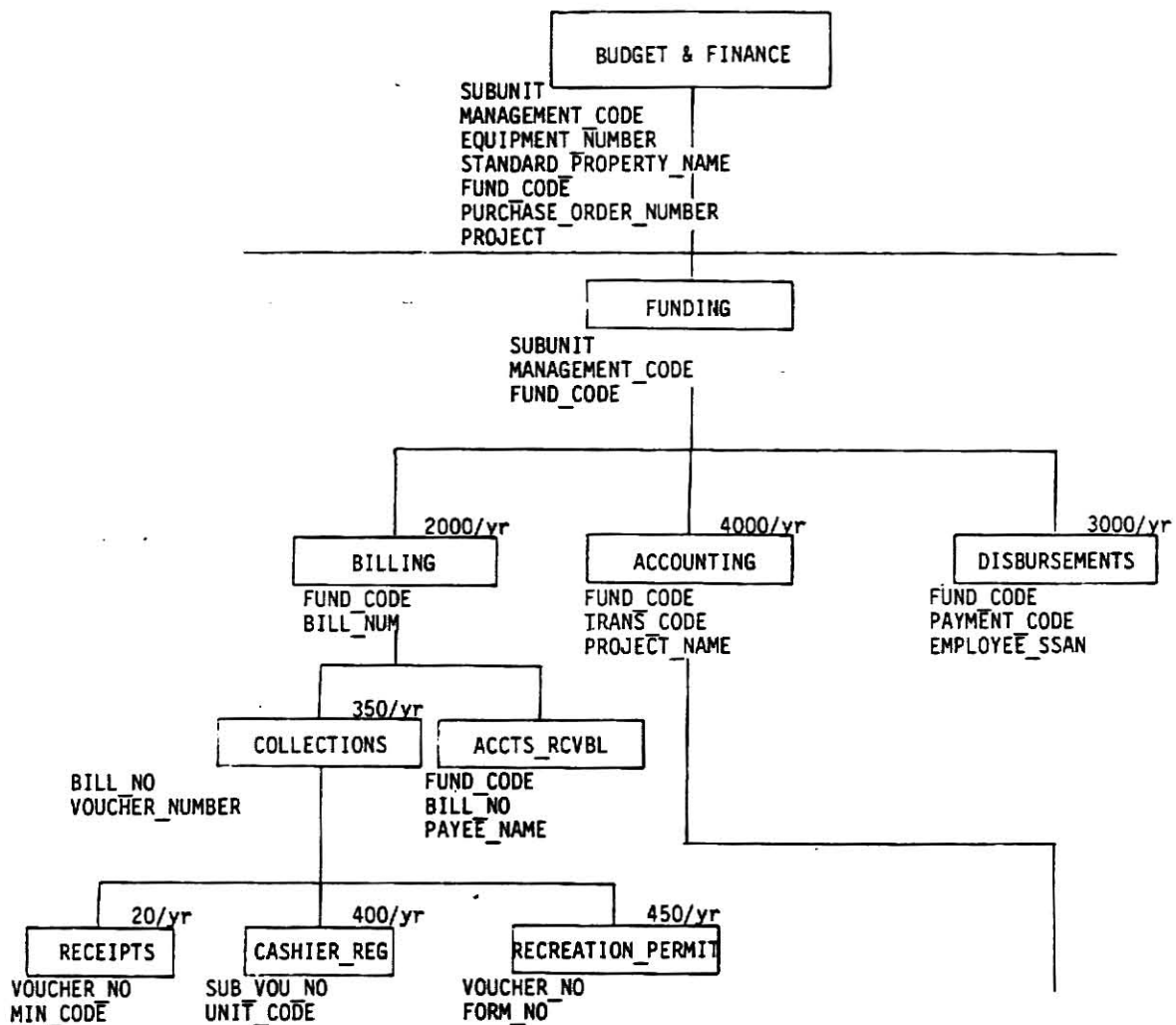


Figure 3A  
 BUDGET & FINANCE  
 FUNDING

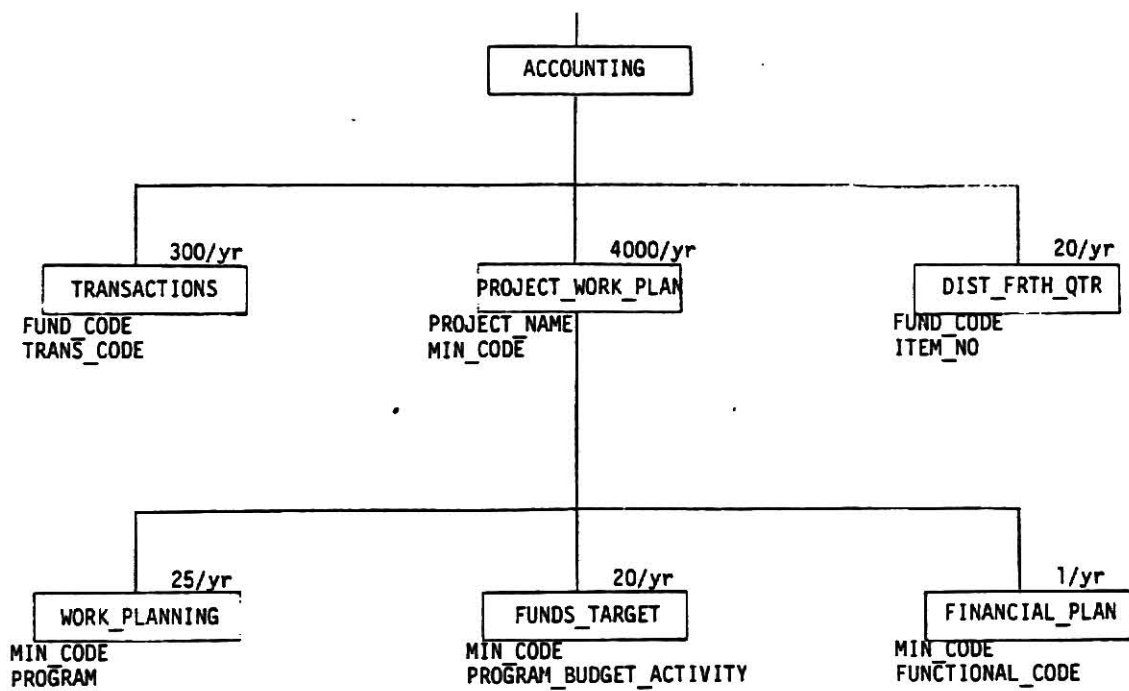


Figure 3B  
BUDGET & FINANCE  
ACCOUNTING

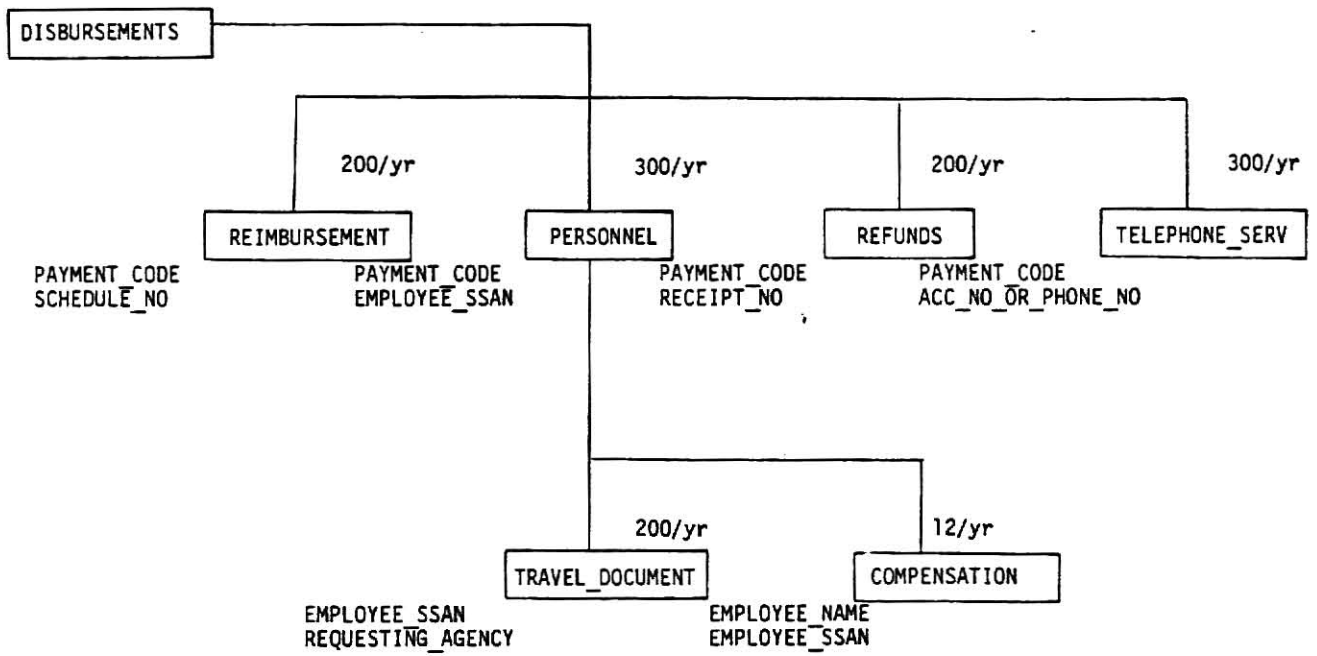


Figure 3C  
BUDGET and FINANCE  
DISBURSEMENTS

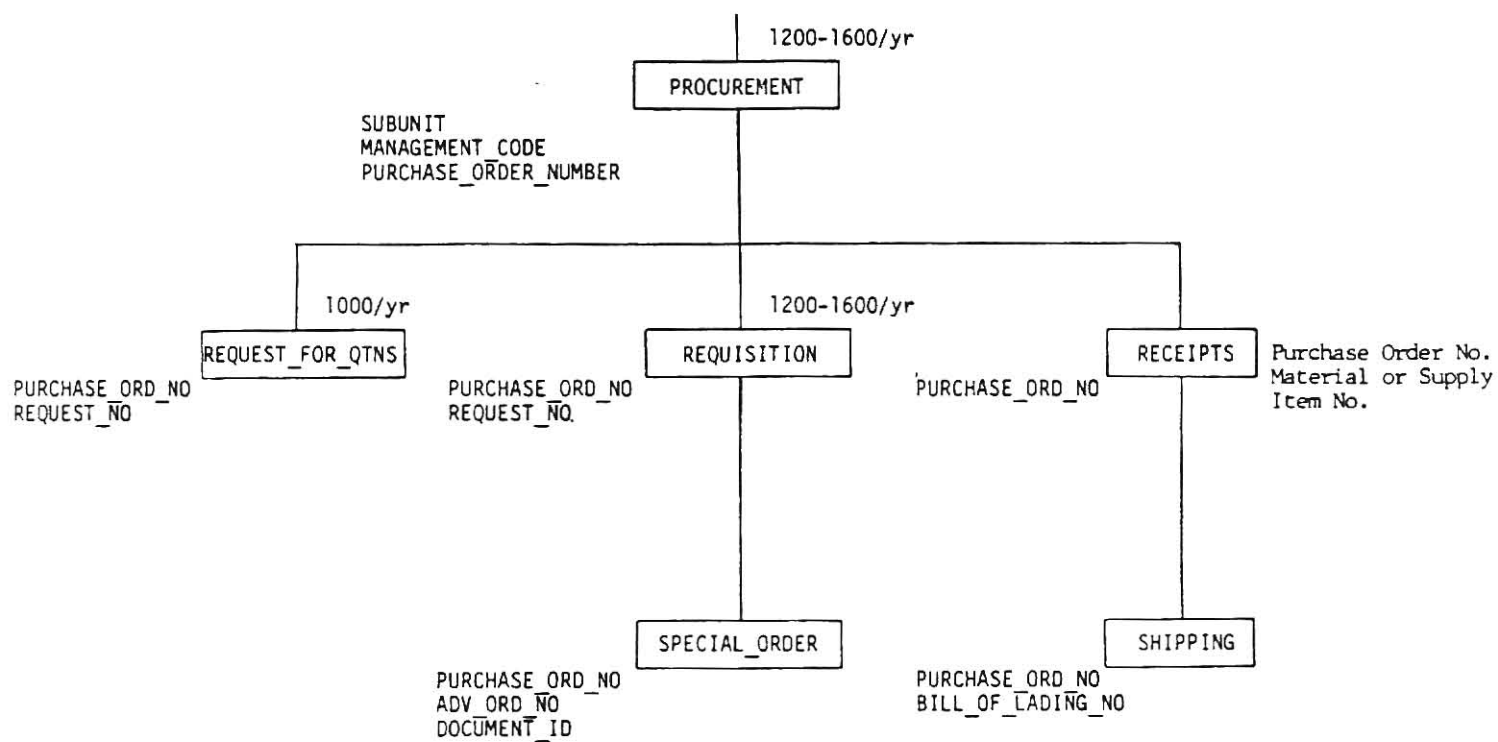


Figure 4  
PROCUREMENT

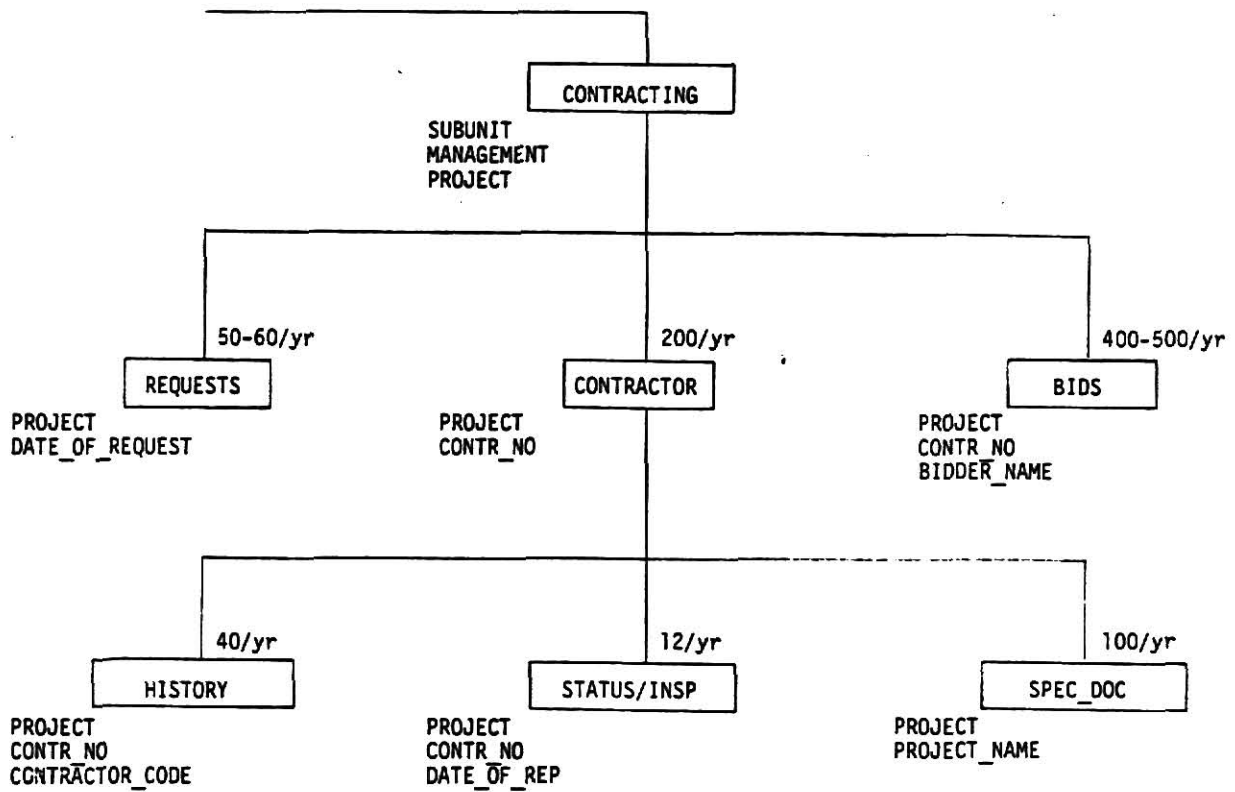


Figure 5  
CONTRACTING

## APPENDIX E

BUDGET / FINANCE NETWORK  
DATA MODEL DIAGRAM

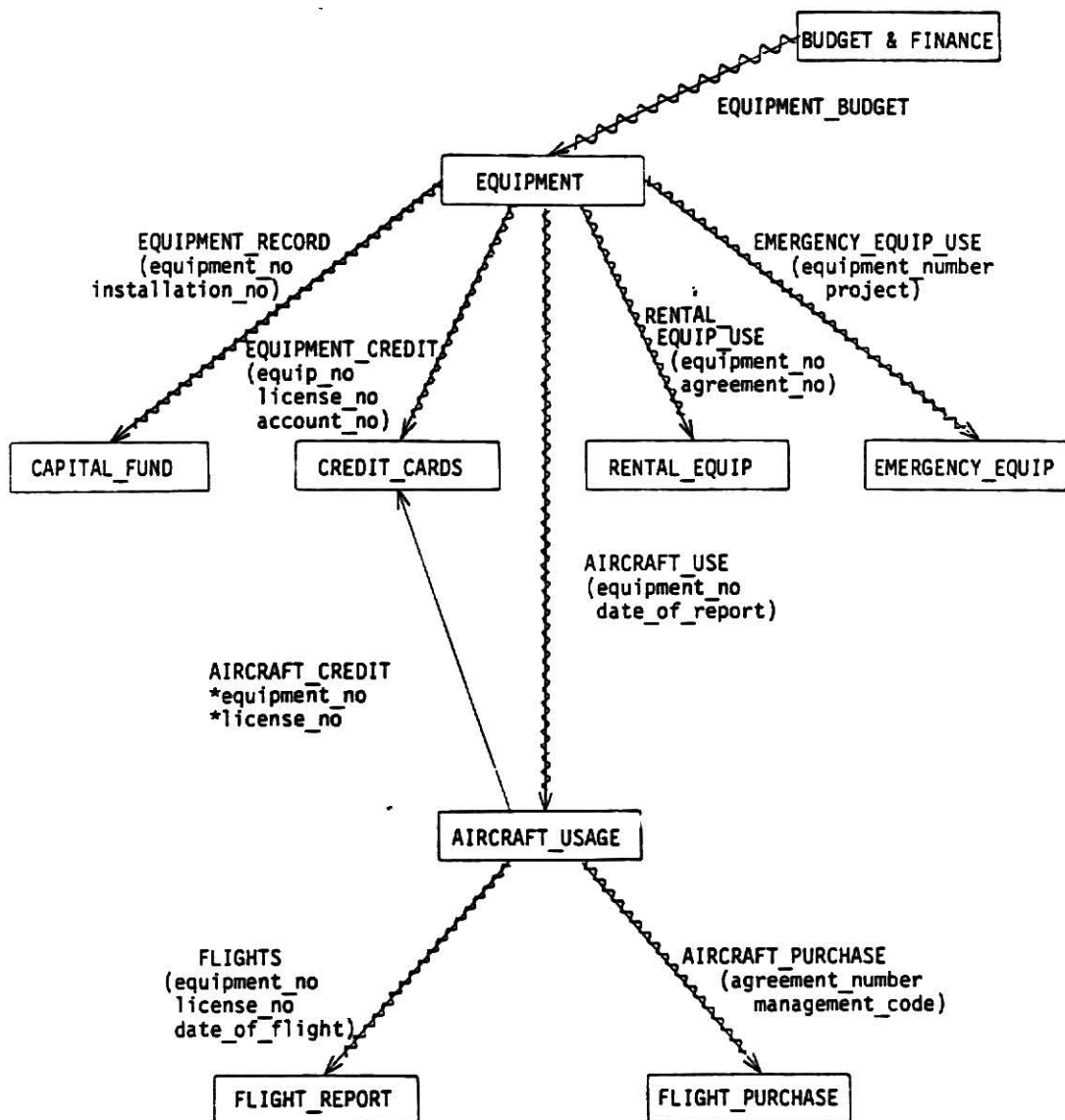


Figure 1  
BUDGET & FINANCE  
EQUIPMENT

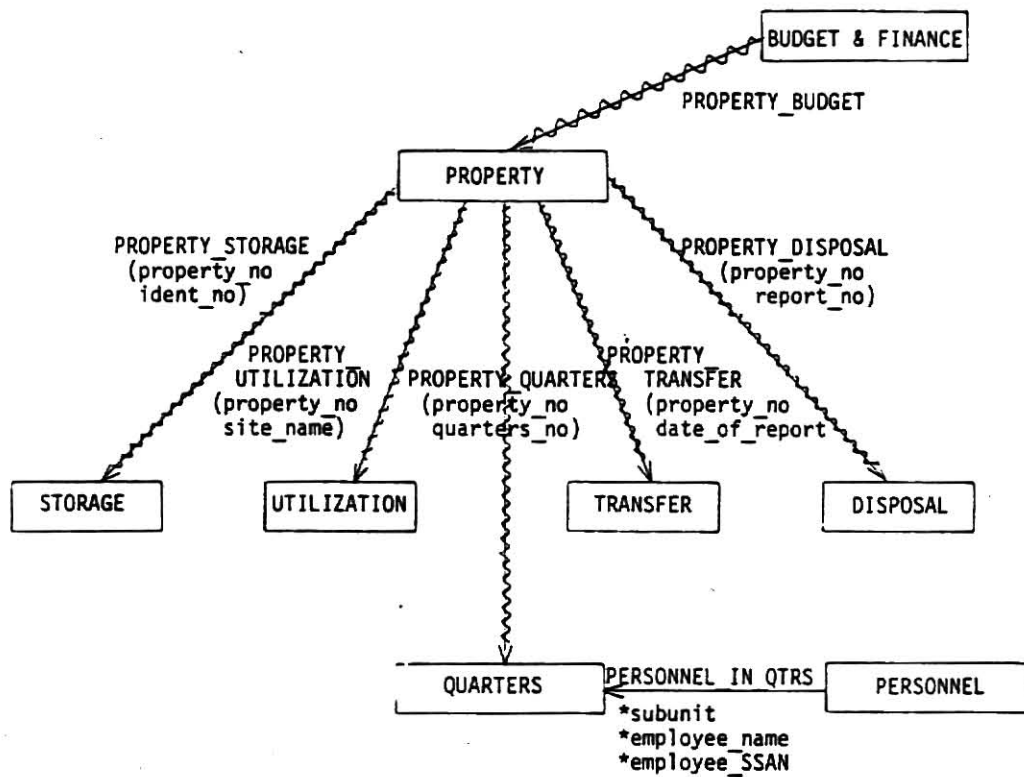


Figure 2  
BUDGET & FINANCE  
PROPERTY



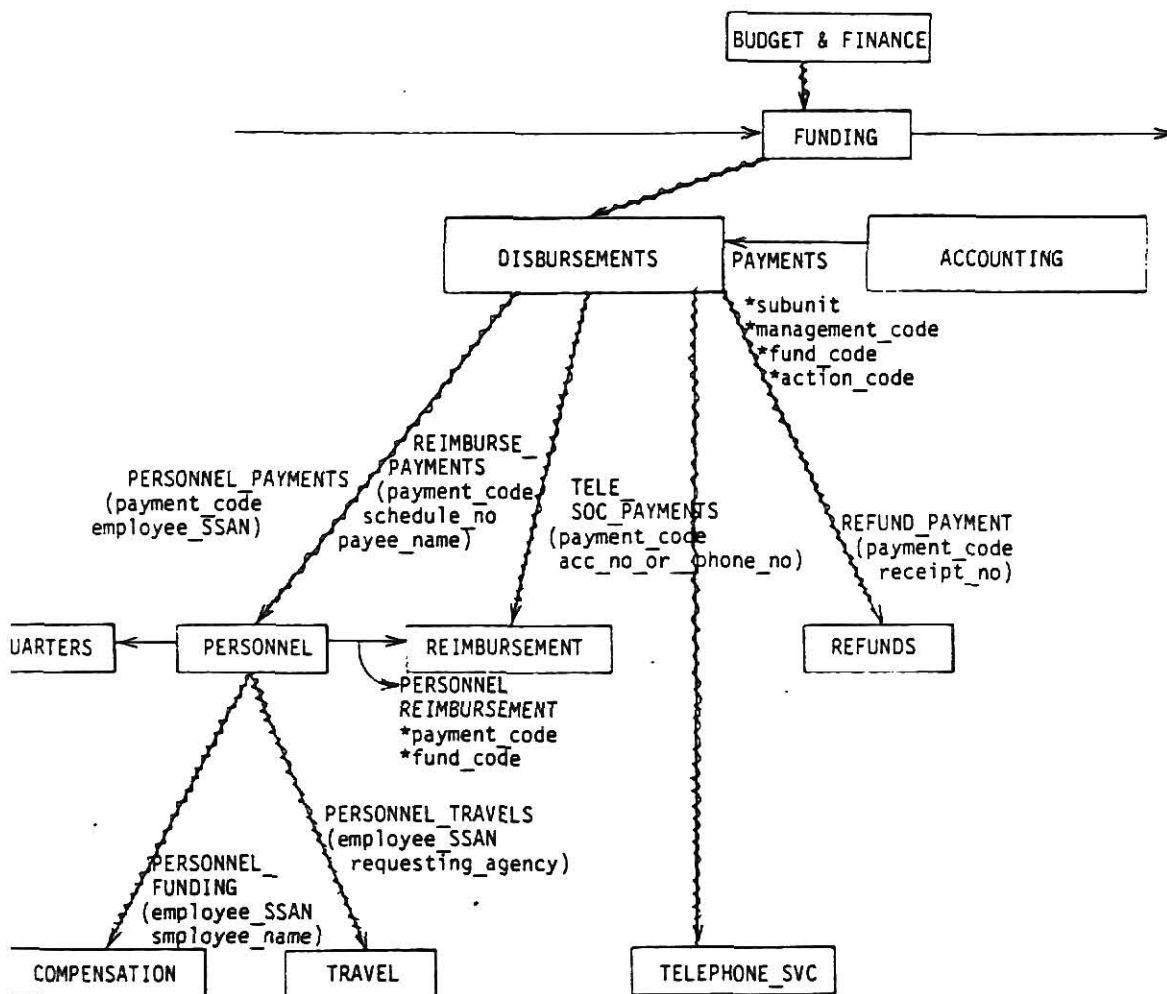


Figure 3B  
FUNDING  
DISBURSEMENTS

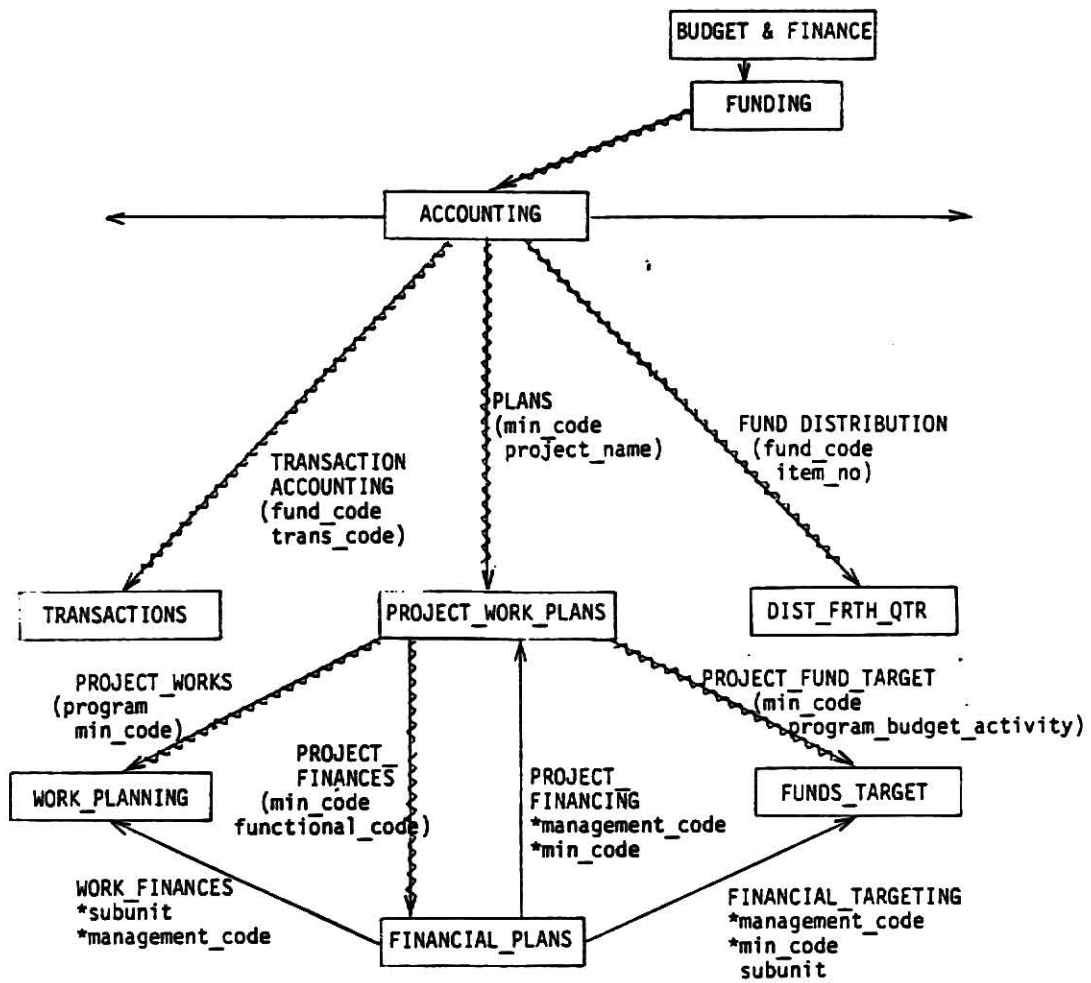


Figure 3C  
FUNDING  
ACCOUNTING

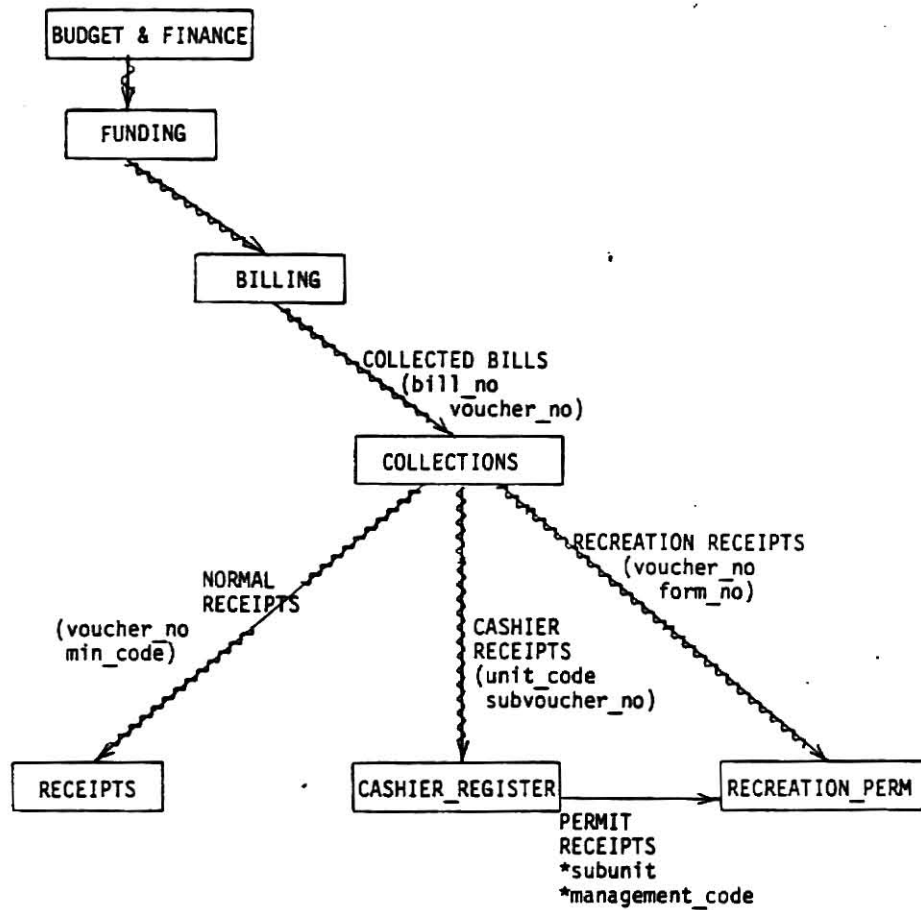


Figure 3D  
FUNDING  
BILLING  
COLLECTIONS

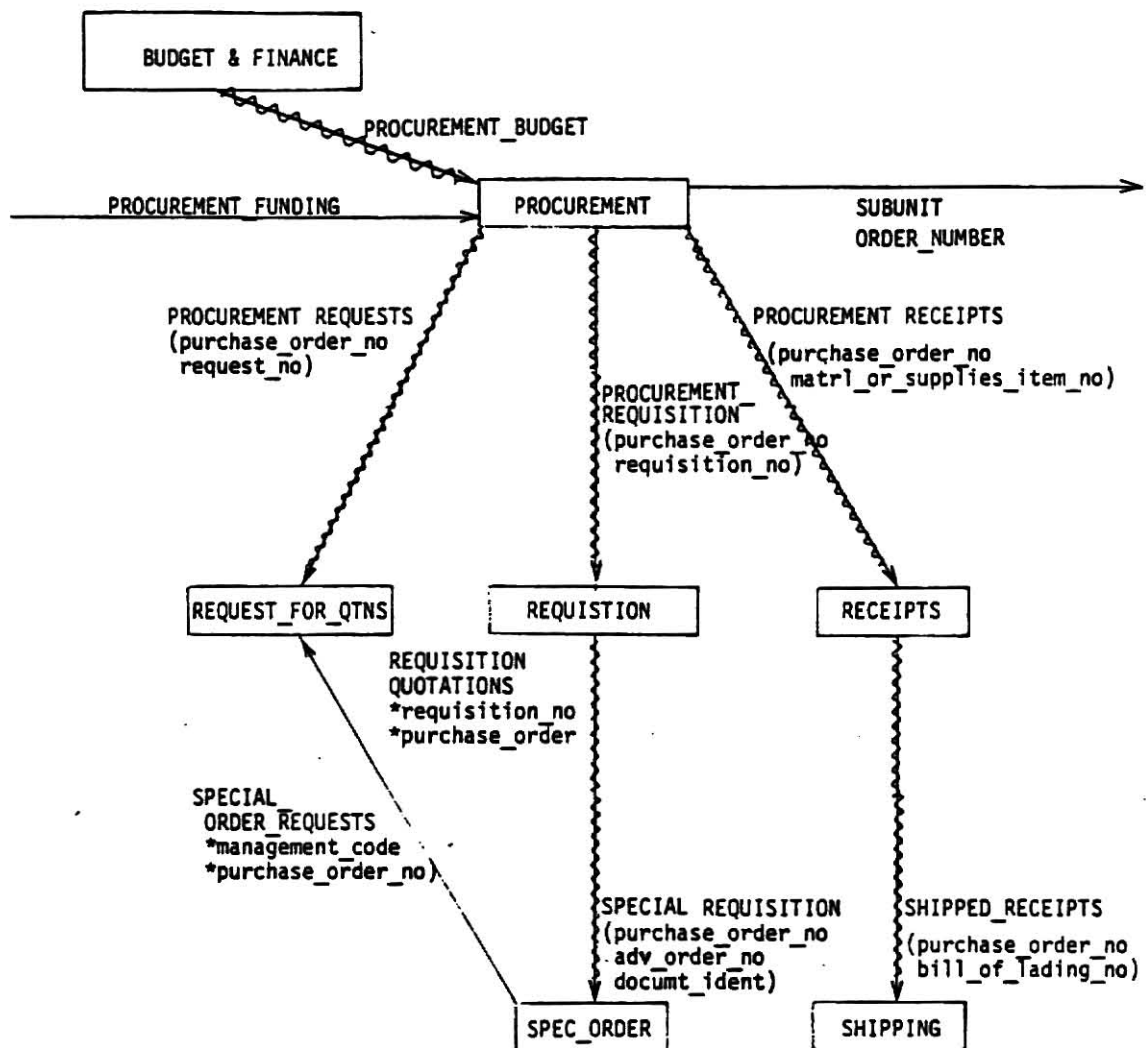


Figure 4  
PROCUREMENT

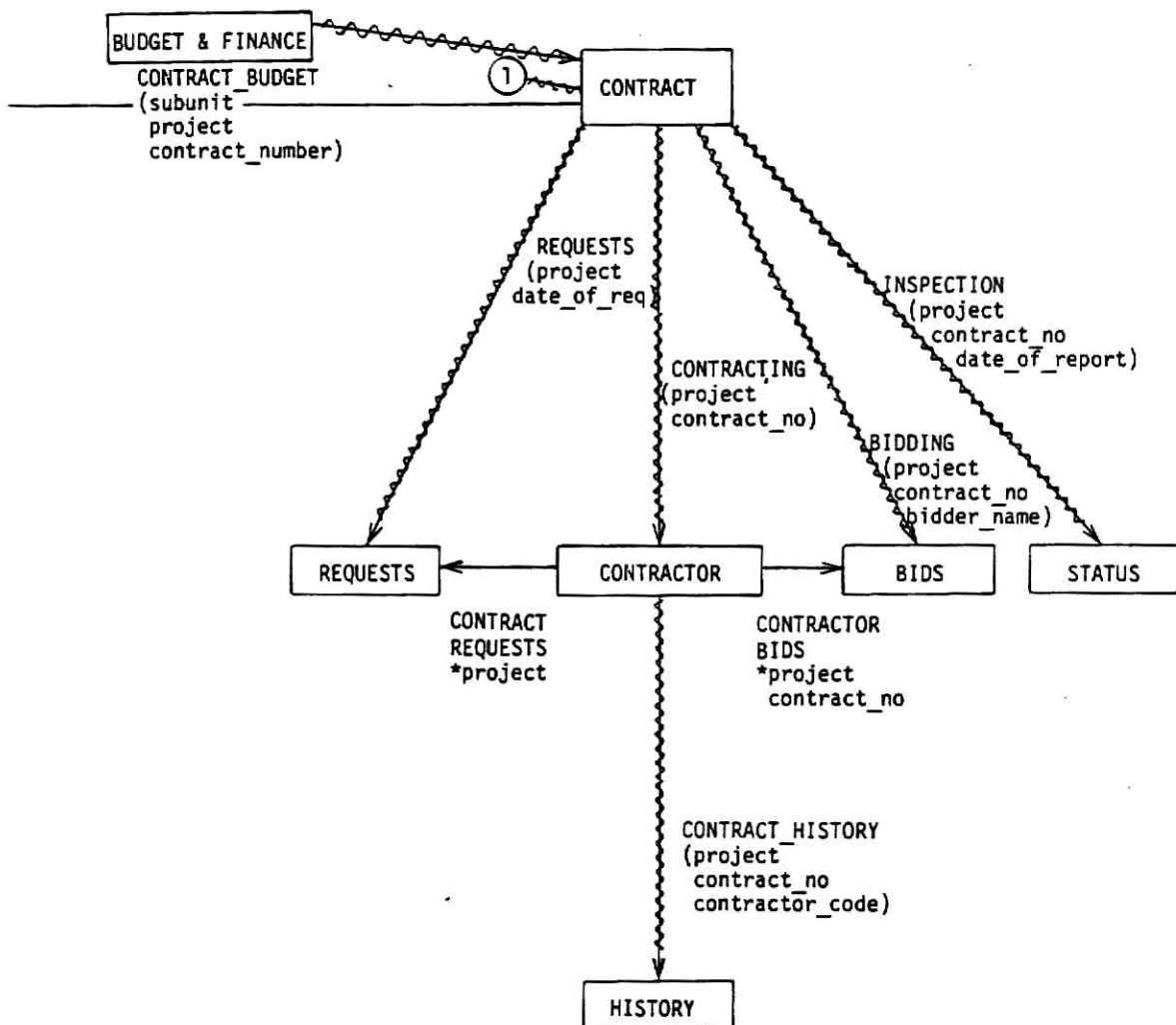


Figure 5  
CONTRACT

DESIGN OF A LARGE DATA BASE  
A METHODOLOGY COMPARISON

by

JAMES R. WILSON

B.S., Brigham Young University, 1970

-----  
AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements of the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY  
Manhattan, Kansas

1983

The design of a large data base presents many challenges to the designer. The many methodologies currently proposed are often confusing to the designer and only tend to compound the problem of determining the proper approach to designing a database.

An integrated approach using aspects of many of the more prominent techniques often is the best answer to this problem. Two approaches earlier proposed by former students are used to design the Budget and Finance section of a database for use by the Ozark National Forest.

A Document Entity Diagram generation system has been proposed which provides an algorithmic approach to the design of a data base using user documents and a graphic manipulation of diagrams. The result of this manipulation is an E-R model which can be used to implement the data base.

Document Handler is an automated approach which also uses the user documents and generates a third normal form relational schema when used in combination with the Bernstein Algorithm Program.

This report will briefly review the more notable design methodologies used in current database design problems. It will discuss the background of the design project and the analysis phase which involves the integration of many of those methodologies. It will then discuss the use of the two approaches in the design process, compare their effectiveness, discuss problems encountered and make recommendations for their use in future design efforts.