ACHIEVING LOGICAL SEPARATION OF THE DECISION SUPPORT SYSTEM DATABASE FROM OPERATIONAL DATABASES

by

D'Ann Naylor Rifai

B.S., Oklahoma State University, 1983
M.B.A., Kansas State University, 1984

A THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988

Approved by:

Major Professor

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

This thesis is dedicated to my husband Imad, who labored for endless hours to produce the fine graphics for this document. In addition, he provided encouragement, support and understanding as I completed this work while employed full time. His patience, confidence in me, and hard work enabled me to achieve the goal of completing my degree. Many thanks are due to my parents and other family members for their encouragement as well.

I would also like to thank my committee members, Dr. Virgil Wallentine, Dr. Richard McBride, and Dr. David Gustafson, for their comments concerning my work. Finally, I express my deep appreciation to my major professor, Dr. Elizabeth Unger, for her patience and willingness to help. She provided many useful suggestions through numerous readings of draft versions of this document.

CHAPTER I. INTRODUCTION

The concept of decision support systems (DSS) originated with Gorry and Morton and their descriptions of futuristic "management decision systems." [Gorr71, p. 55] Recently, a decision support system was defined as an "interactive system that provides the user with easy access to decision models and data in order to support semi-structured and unstructured decision-making tasks." [Wats83, p. 82] Since the introduction of the concept in the early seventies, literature in the area of decision support systems has investigated numerous issues, ranging from the nature of the cognitive process to the methodology for building a generalized decision support system. This paper will focus upon the issue of combining information from multiple data sources in order to meet the needs of decision makers in a specific problem domain.

A.   Decision Support System Components

A DSS is typically considered to be composed of three logically independent subsystems, as shown in Figure I-1 and described below.

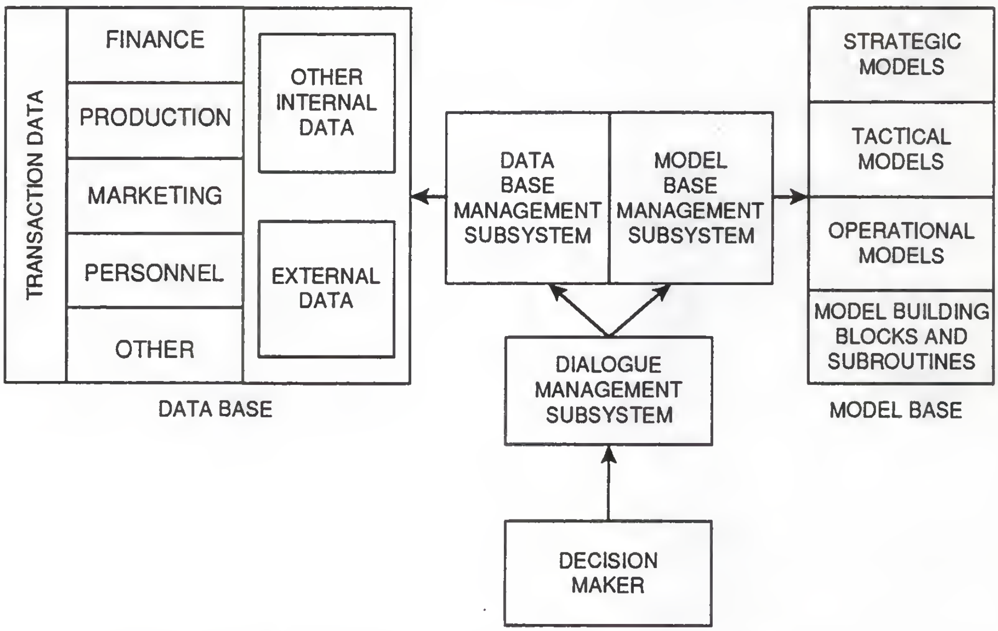


FIG I-1: COMPONENTS OF A DECISION SUPPORT SYSTEM

(adapted from Spra83a, p. 22)

(1) A dialogue management subsystem interfaces with the user and provides transformations between the user's vocabulary and the system's internal modeling and data access vocabulary.

(2) A data management subsystem retrieves data for the user's analysis. The data management subsystem should have traditional DBMS capabilities as well as the ability to interact with other data systems in order to provide comprehensive data for analysis. Information can be retrieved from databases internal to the user's organization, as well as from external data sources such as database systems operated by other organizations. These database systems may contain information concerned with day-to-day operations of the organizations, in addition to summary information used for management reporting. Information from external sources is especially important for decision making at the upper management levels of an organization. Due to the incorporation of external data, "most successful DSS's have found it necessary to create a DSS database which is logically separate from other operational databases." [Spra83b, p. 109]

Both internal and external database systems which provide information for decision making will be referred to as <u>operational databases</u> in this paper.

(3) A model management system is used to apply models to raw data retrieved by the data management subsystem in order to fulfill requests obtained from the user via the dialogue management subsystem. The use of models and their management are the features that distinguish Decision Support Systems from more traditional information processing systems. The capabilities to invoke, run, change, combine and inspect models are key features of this subsystem.

These three subsystems interact to provide the user with data and tools for making decisions. While there are many interesting problems associated with development of a decision support system, this paper will concentrate on the data management subsystem and the issue stated above: a DSS database should be logically separate from other operational databases.

B. Logical Separation of the DSS Database from Other Operational Databases

Sprague's concept of the logical separation of the DSS database from other operational databases means that the data available from the Decision Support System as a whole can be viewed as coming from a single database system. Therefore, the user's view of the structure of the data can be different from the actual structure of the data in the operational databases, as long as the DSS can support the conversion of the data from the operational structures to the user's view. This concept can be implemented in two ways: (1) A physical DSS database can be created that contains data relevant to decision making. This data is replicated from the operational databases. (2) An interface can be created which allows the users to view the DSS database as a single database system, but which actually obtains its data from the operational databases dynamically. Each of these options is described below.

B.1 Creation of a Separate Physical DSS Database

The first option, to create a separate physical DSS database, will allow the DSS to retrieve information from a single database system with little processing overhead. However, this option will present several problems: (1) How often should the information be updated? How recent must the information be for decision-making? Should information to be used for decision making have less stringent requirements for currency? (2) What are the consequences of the inevitable inconsistency if immediate updates to the DSS database are not performed with each operational database update? (3) Is the value of information gained worth the cost of information redundancy? In order to maintain timely data, the cost to maintain the DSS database with updates from numerous databases will be high.

If this option is selected, procedures must be developed to "map" information from the operational database structures to the logical DSS database structure.

B.2 Creation of an Interface which Allows Users to View the DSS Database as a Separate System

The second option is creation of an interface between the different operational databases which allows information to be retrieved dynamically through their DBMSs. In this case, the DSS database is only created in a logical sense. Therefore, the user will view the DSS as retrieving information from a single database system, although information is actually retrieved from several heterogeneous systems. This type of system will require more overhead processing in order to retrieve and reconcile data from different heterogeneous systems, but it will not face the problems and costs of maintaining redundant information.

This option also requires that information be "mapped" from the operational database structures to the logical DSS database structure.

B.3 Distinction between the Operational and Virtual Levels of the DSS

Note that we have distinguished the data which resides in the operational databases from the data which will be available from the DSS as a whole. We will refer to the data from the operational databases as "information at the <u>operational level</u>." The data which will be accessible from the Decision Support System may or may not physically exist in a separate DSS database, depending upon which implementation option described above is selected. Therefore, we will refer to this data as "information at the virtual level." The terms "information at the virtual level" and "information from the DSS database" will be used interchangeably.

B.4 Information Available at the Virtual Level

Theoretically, all of the information available from each of the operational databases can be considered a part of the DSS database. However, for a particular problem domain, all of the operational information may not be of interest for decision making purposes. A DSS Database Administrator (DBA) will determine the information that should be included at the virtual level for the problem domains of interest to the organization developing the DSS.

This paper outlines a framework which can be used by the DSS DBA to specify which operational information is to be available from the Decision Support System, or virtual level. In addition, the framework includes specifications for mappings which translate operational database information to the virtual level.

In order to establish these mappings, information in the different databases must be described using a common data model. According to Hawryszkiewycz, "data model is the accepted term to define the data structure provided at the user interface by a [Database Management System (DBMS)]. Data models are chosen to provide constructs that can model a variety of user problems. ...The goal of any DBMS is to present to the user an interface that emphasizes the logical structure of a user problem and is independent of computer physical structure." [Hawr84, pp. 276-279] The framework proposed in this paper will be based upon a commonly-used data model, the Entity-Relationship Model, which will be described in Chapter III. This paper assumes that the DBA which specifies the mappings from the operational to the virtual level will be familiar with the E-R data model and with terminology for relational database systems.

C. Organization of this Paper

This chapter has served as an introduction to the problem domain for this thesis. Chapter II will provide additional discussion of the problem to be addressed and explain the approach for solving the problem. Chapter III will explain the Entity-Relationship Model, and Chapter IV will illustrate its use for an example problem. Chapter V will outline an extension of the E-R methodology to specify mappings from the operational to the virtual level. Examples will be provided for each step in the process. Chapter VI will summarize the notation used in mappings from the operational to the virtual level. Finally, Chapter VII will state the conclusions of this work and suggest further research for addressing the thesis problem.

CHAPTER II. PROBLEM STATEMENT AND OBJECTIVES

In the previous chapter, we noted that often several different operational databases can contribute useful information for making a decision. However, information about the same physical objects may take different forms in different databases, having different descriptors or attributes, different variable names, and different types of values. Translations may be necessary to combine and reconcile information from the different database systems to form a useful source of data for decision making.

If the DSS database is to be logically separate from the operational databases, and perhaps contain different types of information, then a series of mappings must be established between the operational level, which provides the data, and the virtual level. Although the term "mapping" is often used for other purposes in database literature, in this paper it will be reserved to describe the special case of identifying equivalences so that information at the operational level can be translated into information at the virtual level.

The problem of mapping on a conceptual or semantic level will be addressed by this paper; problems of dealing with different types of Database Management Systems (DBMSs), such as hierarchical, network, and relational data models, will not be addressed. We will assume that technical issues of dealing with different types of database structures can be overcome by translating the data structures into an equivalent form (such as relational). Similarly, language problems can be overcome by transforming the data, if necessary, so that it can be accessed using a common language. (For instance, information from each of the databases can be placed into "flat files" and read using a common language.) The goal of this project is to develop a framework so that data which is semantically equivalent can be identified.

The mappings established using this framework will provide a comprehensive conceptual view of the information available from the DSS database. This comprehensive view will facilitate the interface with the user, enabling him to perceive the DSS as retrieving data from a single DBMS and relieving him of the responsibility for navigating within the different operational databases. The mappings will also provide an internal facility for the DSS, enabling the system to combine and reconcile information from different heterogeneous operational databases. Using these mappings, a natural language interface can be developed, with the dialogue management subsystem interpreting user queries based on the comprehensive conceptual view of the virtual level. The data management subsystem can assume the burden of navigating among the data.

For an example problem situation, three fictitious database systems will be described which will provide information for a problem faced by a government agency. One of the database systems was designed by the government agency, and the other data sources are external database systems. The decision problem is a situation faced by middle management at the agency, one of identifying potentially responsible parties involved in contamination of drinking water supplies. We will focus upon a particular subtask of the decision problem, that of identifying all parties in a waste-handling chain which involves a particular waste type.

The next chapter will review the Entity-Relationship data model and Chapter IV will discuss how it is used for representing information in database systems. Chapter V will show the steps for specifying mappings from the operational to the virtual level.

CHAPTER III. METHODOLOGY

This chapter will describe the Entity-Relationship data model and illustrate how the model is used to represent information in database systems. Peter Chen originally proposed the Entity-Relationship (E-R) Model in 1976 as a generalization of several existing models for representing the logical view of data. The Entity-Relationship model adopts the view that the real world consists of entities and relationships. The terminology for Chen's E-R Model will be presented below.

A. Entities, Relationships, and Attributes

An entity is defined as a "'thing' which can be distinctly identified. A specific person, company, or event is an example of an entity." [Chen76, p. 10] Entities are represented using the notation E_i and are classified into entity sets such as PERSON or COMPANY.

A relationship is defined as "an association among entities. For instance, 'father-son' is a relationship between two 'person' entities." [Chen76, p. 10] Relations are classified into relationship sets. "A relationship set, R_i, is a mathematical relation among n entities, each taken from an entity set:

$$\{[e_1, e_2, \ldots, e_n] \mid e_1 \varepsilon E_1, e_2 \varepsilon E_2, \ldots, e_n \varepsilon E_n\},$$

and each tuple of entities, $[e_1, e_2, \ldots, e_n]$ is a relationship." [Chen76, pp. 11-12] Because Chen's notation specifies each entity involved in a relationship, we will follow this notation and assume that only whole entities can be involved in a relationship.
Chen states that "the information about an entity or a relationship is obtained by observation or measurement, and is expressed by a set of attribute-value pairs. '3', 'red', and 'Peter' are values. Values are classified into different value sets, such as FEET, COLOR, FIRST-NAME and LAST-NAME. ...An attribute can be formally defined as a function which maps from an entity set or a relationship set into a value set or a Cartesian product of value sets:

$$f: E_i \text{ or } R_i \rightarrow V_i \text{ or } V_{i1} \times V_{i2} \times \ldots \times V_{in}." \quad \text{[Chen76, p. 12]}$$

In an E-R diagram, an entity is represented as a rectangle, a relationship is represented as a diamond, and an attribute is represented as a circle which emanates from the entity or relationship which it describes. A simple E-R diagram is presented in Figure III-1 below.

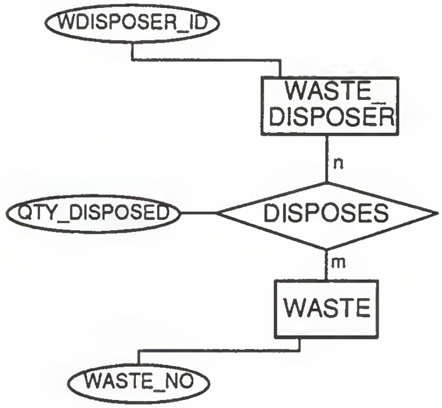


FIG. III-1: A SIMPLE E-R DIAGRAM.

This entity-relationship diagram depicts the WASTE_DISPOSER entity set, which represents companies which dispose of hazardous waste; the WASTE entity set, which represents hazardous wastes and their characteristics; and the DISPOSES relationship, which is an association between WASTE DISPOSER companies and WASTE streams. This example contains a subset of the information that will be used in the example in Chapter IV.

It is necessary to uniquely identify each entity in an entity set. Therefore, one or more entity attributes are used as an entity identifier; these attributes are known as key attributes. Because it is possible that more than one set of attributes uniquely identifies an entity, one set of key attributes will be specified as the primary key.

Similarly, relationships must be uniquely identified within a relationship set. Usually the key for a relationship set includes the key attributes of the entities that participate in the relationship. While non-key attributes of the related entities are NOT duplicated in the relationship set, the relationship may have attributes of its own which are included.

Data for entities and relationships can be represented as files of records, tables of data, relations of tuples, or in other ways. Three tables of data are used to represent the entities and relationships from Figure III-1; one contains information about waste disposal companies, one has information about wastes, and the third has information about the relationship. In the tables representing entities, each row of values is related to the same entity, and each column represents an attribute of the entity. In the table representing the relationship between the two entity sets, only attributes which uniquely identify the entities being related are shown.

WDISPOSER_ID	OTHER DATA
10	
20	
30	
40	
50	

Table III-1(a) WASTE DISPOSER data

WASTE_NO	OTHER DATA
160	
170	
180	
190	

Table III-1(b) WASTE data

WDISPOSER_ID	WASTE_NO
10	190
30	160
30	180
40	180
40	190

Table III-1(c) DISPOSES relationship data

TABLE III-1: TABLES OF DATA FOR THE E-R DIAGRAM OF FIGURE III-1

The next section will describe how information in an E-R model is converted to a relational database.

B. Conversion from the E-R Model to a Relational Database

The representation of the data for the WASTE DISPOSER and WASTE entities in tables is natural and easy to understand. According to Date [Date86, p. 96], a relational database is a database that is perceived by its users as a collection of tables. Each row of a table is called a tuple and represents an entity or relationship, and each column of a table represents an attribute.

According to Hawryszkiwycz [Hawr84, pp. 117-120], usually an E-R Model is converted to relations by converting each entity set and each relationship set to a relation. "The attributes of entities in the entity set become the attributes of the relation, which represents that entity set. The entity identifier becomes the key of the relation. Each entity is represented by a tuple in the relation. Similarly, the attributes of relationships in each relationship set become the attributes of the relation, which represents the relationship set. The relationship identifiers become the key of the relation. Each relationship is represented by a tuple in that relation." [Hawr84, p. 117] We will refer to these two types of relations as <u>entity relations</u> and <u>relationship relations</u>.

C. The Essence of the Problem

Chen points out that "some people may view something (e.g. marriage) as an entity while other people may view it as a relationship. We think that this is a decision which has to be made by the enterprise administrator. He should define what are entities and what are relationships so that the distinction is suitable for his environment." [Chen 76, p. 10] Different DBAs and different situations may cause a shift of entities to relationships, and vice versa.

In the DSS environment, information is obtained from multiple heterogeneous databases. Therefore, it is possible that an entity from one database can be represented as any of the following in a second database: an entity, a relationship, an attribute of another entity, or an attribute of a relationship. This fact presents a problem when combining information from these different databases and requires that we establish a mapping between different representations of the same physical object.

Chapter IV will present an example problem situation in which different databases describe the same physical entities in different ways.

CHAPTER IV. THE EXAMPLE DECISION PROBLEM

As noted earlier, the Decision Support System environment often requires that information be obtained from multiple heterogeneous database systems. This fact presents a problem, because information from these different systems that describes the same physical object is often structured in different ways. Consequently, mappings must be provided to identify which data components are semantically equivalent.

In this chapter, an example decision problem illustrating this problem will be given. This example problem will be used in Chapter V to derive a virtual representation of the information of interest for the problem situation.

A. Statement of the Decision Problem

Assume that the government has just discovered an extremely hazardous substance in the drinking water supplies for a metropolitan area. The government is forced to immediately finance a multi-million dollar clean-up effort to prevent danger to the inhabitants of the area.

By law, the government is allowed to recover its clean-up costs from all responsible parties that contributed to the hazardous situation. A complex modeling program will be used to determine possible sources for the substance, given its nature, quantity and direction of movement. Parties which must assume responsibility include not only the direct source of the waste, but also other parties which have sent hazardous material to the direct source. For instance, if the substance is leaking into the groundwater from a municipal landfill, all companies which send waste to the landfill for dumping are potentially liable. These situations are common in the hazardous waste industry; companies which collect waste from other companies handle large volumes of waste and have more potential for contamination problems. A search must be conducted to identify all potentially responsible parties for the hazardous substance.

B. The Relevant Databases

No single database exists which can provide all of the information needed to address the problem of identifying responsible parties. However, the government does have access to three national databases: (1) a National Manufacturing Organization (NMO) database which includes data about by-products produced as a result of manufacturing and the companies which collect those by-products; (2) a National Shipping Organization (NSO) database which includes information about yearly shipments of particular materials; and (3) an Environmental Protection Agency (EPA) database used for regulating facilities which process or dispose of hazardous waste.

Naturally, none of these databases is tailored to the problem of identifying responsible parties; each has been designed for its own purpose. However, because of national government requirements for reporting transactions involving hazardous materials, each of the national databases contains information applicable to the problem situation and solution.

B.1 The National Manufacturing Organization Database

For example, the NMO's primary purpose is to identify manufacturers of different products. The database contains information about each manufacturer, in addition to the number of units of each product type produced yearly. Because of government reporting requirements, the database also contains information concerning quantities of by-products produced. As a service to its clients, the NMO also provides information concerning organizations which handle the hazardous materials and yearly quantities transferred from each manufacturer to each recipient.

As shown in Figure IV-1, entities for this database include MANUFACTURERS, PRODUCTS, BY_PRODUCTS, and RECIPIENTS. Only those attributes of each entity which are involved in the example decision problem are illustrated.

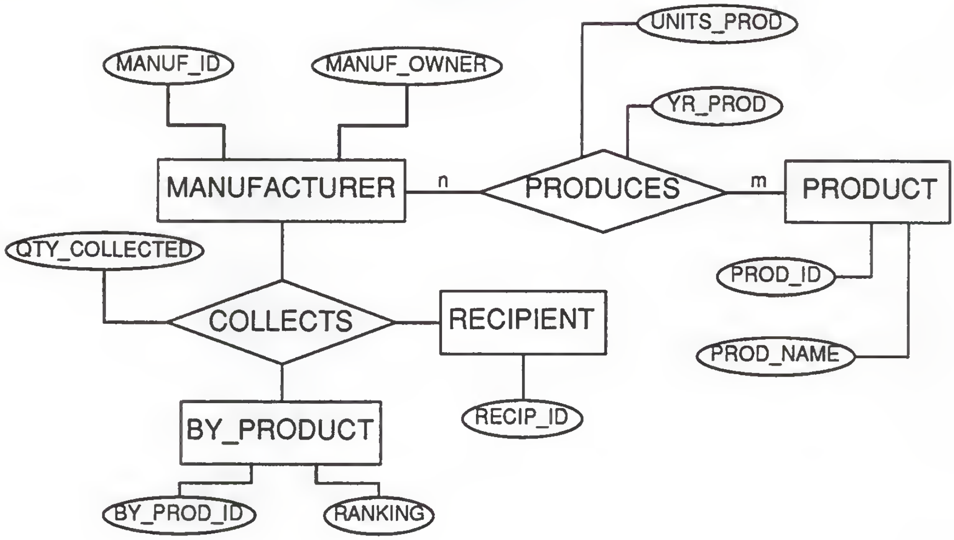


FIG. IV-1: E - R_1
OPERATIONAL DATABASE 1

NATIONAL MANUFACTURING ORGANIZATION'S DATABASE

Note that, in order to trace the transfer of a particular BY-PRODUCT from a MANUFACTURER to a RECIPIENT, one relationship is used to associate the three entities. An assumption made in using one relationship to associate the three entities is that all BY-PRODUCTS will be transferred from a MANUFACTURER to a RECIPIENT; if it is necessary to represent that a MANUFACTURER actually retains the BY-PRODUCT on-site, a dummy RECIPIENT record would be created to represent the MANUFACTURER as a RECIPIENT.

Data representing the MANUFACTURER, BY_PRODUCT, and RECIPIENT entities and the COLLECTS relationship are presented below.

MANUF_ID	MANUF_OWNER	OTHER DATA
M050	A Corp.	
M100	Joe Smith	
M150	Z Corp.	
M200	XYZ Corp.	
M250	Sam Jones	

Table IV-1(a): MANUFACTURER data

RECIP_ID	OTHER DATA
SC05	
SC10	
SC20	

Table IV-1(b): RECIPIENT data

BY_PROD_ID	OTHER_DATA
BP001	
BP002	
BP003	
BP004	

Table IV-1(c): BY_PRODUCT data

MANUF_ID	BY_PROD_ID	RECIP_ID	QTY_COLLECTED
M050	BP002	SC05	114
M100	BP004	SC20	10
M250	BP004	SC20	25
M200	BP001	SC10	50

Table IV-1(d): COLLECTS relationship data

TABLE IV-1: TABLES OF DATA FOR THE NATIONAL MANUFACTURING ORGANIZATION'S DATABASE

B.2 The National Shipping Organization Database

The NSO traces amounts of materials shipped yearly from each pick-up source to each destination. Entities for this database include SHIPPING COMPANIES and SHIPMENTS, as illustrated in Figure IV-2 below. Only those attributes of each entity which are involved in the example decision problem are illustrated.

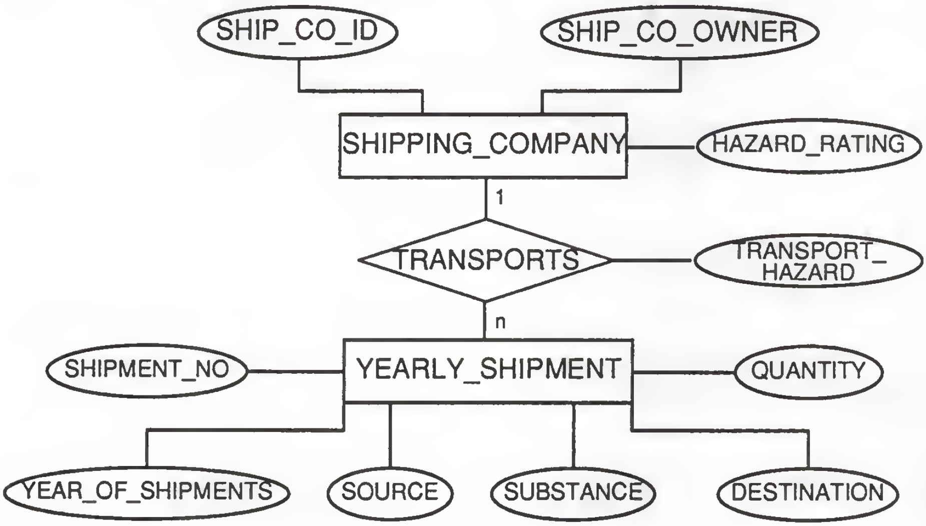


FIG. IV-2: E - R_2
OPERATIONAL DATABASE 2

NATIONAL SHIPPING ORGANIZATION'S DATABASE

Note that the SOURCE of a shipment and its DESTINATION are given as attributes of a SHIPMENT; they are not of sufficient importance to the national organization to be considered independent entities. Data representing the SHIPPING_COMPANY and YEARLY_SHIPMENTS entities and the TRANSPORTS relationship are shown below.

SHIP_CO_ID	SHIP_CO_OWNER	OTHER DATA
SC05	A Corp.	
SC10	ABC Corp.	
SC20	XYZ Corp.	

Table IV-2(a): SHIPPING_COMPANY data

SHIP_CO_ID	SHIPMENT_NO	TRANSPORT_HAZARD
SC05	30	.75
SC05	40	.8
SC20	10	1
SC20	20	.95
SC10	50	.55
SC10	60	.45

Table IV-2(b): TRANSPORTS relationship data

SHIPMENT_NO	YEAR_OF_SHIPMENTS	SOURCE	SUBSTANCE	DESTINATION	QUANTITY
10	88	M100	160	WD30	10
20	88	M250	160	WD30	25
30	88	M050	190	WD40	15
40	88	M050	190	WD10	99
50	88	M200	180	WD40	25
60	88	M200	180	WD30	25

Table IV-2(c): YEARLY_SHIPMENT data

TABLE IV-2: TABLES OF DATA FOR THE NATIONAL SHIPPING ORGANIZATION'S DATABASE

The TRANSPORTS relationship has its own attribute, TRANSPORT_HAZARD. This attribute represents the potential hazard posed by the shipment;

the hazard depends upon the qualifications of the shipping company (from the HAZARD_RATING attribute) as well as the toxicity of the waste being shipped.

B.3 The Environmental Protection Agency Database

The EPA is mostly concerned with the facilities which actually process and dispose of hazardous waste; these facilities must obtain permits to operate. Consequently, WASTE DISPOSERS, WASTES, and PERMITS are entities in this database, as shown in Figure IV-3. Only those attributes of each entity which are involved in the example decision problem are illustrated.

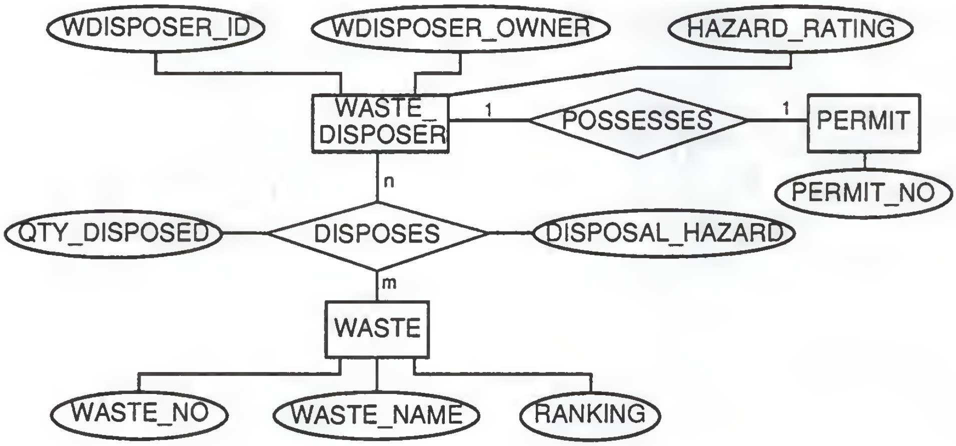


FIG. IV-3: E - R_3
OPERATIONAL DATABASE 3

EPA'S DATABASE OF HAZARDOUS WASTE DISPOSERS

Note that this database contains no information about the origin of a waste; the transfer of waste material to a WASTE DISPOSER is not traced by the EPA. Data representing the WASTE DISPOSER and WASTE entities and the DISPOSES relationship are shown below.

WDISPOSER_ID	WDISPOSER_OWNER	OTHER DATA
WD10	A Corp.	
WD20	B Corp.	
WD30	C Corp.	
WD40	D Corp.	
WD50	E Corp.	

Table IV-3(a): WASTE DISPOSER data

WASTE_NO	WASTE_NAME	OTHER DATA
160	Arsènic	
170	Chromium	
180	Cyanide	
190	Xylene	

Table IV-3(b): WASTE data

WDISPOSER_ID	WASTE_NO	QTY_DISPOSED	DISPOSAL_HAZARD
WD10	190	99	.8
WD30	160	35	1
WD30	180	25	.5
WD40	180	25	.75
WD40	190	15	.8

Table IV-3(c): DISPOSES relationship data

TABLE IV-3: TABLES OF DATA FOR THE ENVIRONMENTAL PROTECTION AGENCY'S DATABASE

The DISPOSES relationship has an attribute similar to the TRANSPORT_HAZARD attribute of the TRANSPORTS relationship. This attribute represents the potential hazard posed by the disposal; the hazard depends upon the qualifications of the disposal company (from the HAZARD_RATING attribute) as well as the toxicity of the waste being disposed (from the RANKING attribute).

C. Information Required from the Decision Support System

In order to solve the problem of tracing a waste from its origin to its final destination, information in the three databases must be combined. We will assume that all WASTES produced are BY-PRODUCTS of manufacturing processes and are shipped via a SHIPPING COMPANY to a WASTE DISPOSER facility. The DSS will be used to connect the links between the different parties handling the particular type of waste causing the immediate problem. Assume that the search can be limited to the geographic area of interest, although this aspect of the problem will not be illustrated. In order to accomplish this task, mappings between the different databases must be provided. Chapter V will provide the framework for specifying the mappings and illustrate development of these specifications using the example databases.

CHAPTER V. THE PROBLEM SOLUTION

Information from sources external to an organization is frequently used in decision-making. Consequently, combining information from different sources is a prevalent problem for Decision Support Systems. This chapter will discuss the proposed problem solution, which has been developed using the Entity-Relationship model, and will provide illustrations of each step in the process using the example databases introduced in Chapter IV.

A. REVIEW OF THE VIRTUAL LEVEL CONCEPT

Recall that the data which resides in the operational databases has been distinguished from the data which will be available through the DSS from the virtual database. The data from the operational databases is referred to as "information at the operational level" and data accessible from the DSS is referred to as "information at the virtual level". Information at the virtual level will probably not include all of the information from each operational database, but it will include all information relevant to the problem domain of the DSS.

Because information about the same physical object may take different forms in different databases, translations may be necessary to combine and reconcile information from the different operational database systems. Therefore, the notion of a mapping has been introduced to identify data which describes the same physical object. Mappings are used at the operational level to identify semantic equivalences in the different operational databases, as well as equivalences between the operational level and the virtual level.

In order to establish mappings between the different databases, the information contained in each of the databases must be described using a common framework. We have chosen to describe each of the databases using the Entity-Relationship (E-R) model. Using the E-R diagrams for the operational databases, mappings between the different components of the databases will be identified to obtain an overall picture of the information available to the DSS. The DBA will use these mappings to evaluate the relevance of each information item to the problem domain and to determine the information that will be available from the DSS. Finally, the framework for specifying mappings from the operational to the virtual level will be presented.

As mentioned in Chapter II, the mappings established using this framework will provide a comprehensive conceptual view of the information available from the DSS database. This comprehensive view will facilitate the interface with the user, enabling him to perceive the DSS as retrieving data from a single DBMS and relieving him of the responsibility for navigating within the different operational databases. The mappings will also provide an internal facility for the DSS, enabling the system to combine and reconcile information from different heterogeneous operational databases. Using these mappings, a natural language interface can be developed, with the dialogue management subsystem interpreting user queries based on the comprehensive conceptual view of the virtual level. The data management subsystem can assume the burden of navigating among the databases to retrieve the relevant data.

B. STEPS IN DETERMINING A VIRTUAL ENTITY-RELATIONSHIP MODEL

Assume there are E-R diagrams for each of the operational databases or that they can be easily derived. We will then go through a four-step process to create and refine the virtual Entity-Relationship model, as illustrated in Figure V-1 below.

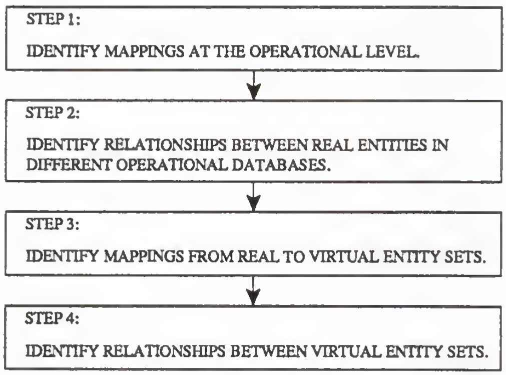


FIG. V-1: STEPS IN DEVELOPING THE VIRTUAL ENTITY-RELATIONSHIP MODEL

At this point, we will clarify the difference between a mapping and a relationship. A mapping will establish an equivalence between two or more entity sets at the operational level or between a real and a virtual entity set. A relationship, however, denotes an association between two or more members of entity sets, or entities, at the same level; either both entities are at the operational level or both are at the virtual level. Relationships cannot cross the operational-to-virtual boundary, but mappings can. In addition, as specified in Chen's model, relationships involve whole entities; mappings may involve only attributes of entities.

In the four-step procedure for developing the virtual Entity-Relationship model, the purpose of the first step is to identify semantic equivalences between the entities and attributes of the operational E-R diagrams. The step which identifies relationships between real entities in different operational databases allows the

DBA to express semantic relationships which are not otherwise represented. Usually these relationships will be the result of a mapping from the first step. For these relationships to become a part of the DSS, the DBA must specify a procedure for determining which entities are involved in the relationships; this information must be derived from the existing real relationships using the mappings specified between entities and attributes at the operational level. The final two steps identify the result of the analysis: the entities that will be mapped to the virtual level and the relationships between them.

The next section will provide the notation to be used to specify the real entities and relationships. Following that, we will provide the notation for specifying the product of this analysis, the virtual E-R. Each subsequent section will discuss the steps in developing the virtual E-R.

B.1 NOTATION FOR THE GIVEN REAL ENTITIES AND RELATIONSHIPS

Basically we describe an entity E as a set of attributes A_i, i=1,n and as having a primary key attribute (or set of attributes) A^k. For the sake of simplicity, throughout this paper the key will be denoted as a single attribute A^k, recognizing that actually several attributes may make up the primary key of a relation.

Within our system, we must be able to distinguish the operational database, or E-R, in which the entity resides. Therefore, a real entity E_{ij} is described by a subscript i which defines the E-R i of its origin and a subscript j which identifies the entity j within the E-R i. When referring to an entity j within a particular operational database i, the entity will be described as "the entity j within the E-R i." Each entity attribute A_{ijk} must then be described using three subscripts, using the same two- subscript notation to identify the entity and a third subscript to indicate the attribute of the entity. The entity's primary key will be described as $A^k{}_{ijl}$, using the three-subscript notation and the superscript k to indicate that the attribute is key.

From our example, the real entities from the operational databases and their key values are listed below.

Operational database	Entity		Primary key	
NATIONAL MANUFACTURING ORGANIZATION $E\text{-}R_1$	MANUFACTURER PRODUCT BY_PRODUCT RECIPIENT	E_{11} E_{12} E_{13} E_{14}	MANUF_ID PROD_ID BY_PROD_ID RECIP_ID	A^k_{111} A^k_{121} A^k_{131} A^k_{141}
NATIONAL SHIPPING ORGANIZATION $E\text{-}R_2$	SHIPPING_COMPANY YEARLY_SHIPMENTS	E_{21} E_{22}	SHIP_CO_ID SHIPMENT_NO	A^k_{211} A^k_{221}
EPA $E\text{-}R_3$	WASTE DISPOSER WASTE PERMIT	E_{31} E_{32} E_{33}	WDISPOSER_ID WASTE_NO PERMIT_NO	A^k_{311} A^k_{321} A^k_{331}

TABLE V-1: REAL ENTITIES IN EXAMPLE DATABASES

A real relationship is a relationship between two entities in the same operational database. Each real relationship R_{ij} is described by a subscript i indicating the real E-R that contains the relationship and a second subscript j which identifies the relationship within the E-R i. Since a real relationship is between entities in the same E-R, all entities in the real relationship will have the same initial subscript.

A real relationship scheme will consist of the keys of the entities being related, in addition to any relationship attributes which are unique to the relationship itself. Because the Entity-Relationship Model allows more than two entities to be associated in a relationship, our notation will allow more than two entities to be specified. Not all relationships will have attributes, and therefore this listing of relationship attributes is optional. Relationship attributes will be specified using the notation A^r_{ijk}, with the third subscript identifying the relationship attribute within the relation and the superscript r indicating that the attribute is unique to the relationship.

The general form for a relationship scheme will consists of the real keys showing the entities being related, followed by the optional relationship attribute set, as given below:

$$R_{ij}\ (\{k_1,\ \ldots k_n\}\ |\ n\geq 2;\ \{RA\}).$$

where n indicates the number of entities that are related;
$k_x \in \{A^k_{ipq},\ p=1,e;\ q=1,f\}$; all keys of the related entities are listed;
i indicates the operational database, or E-R; all entities in a real relationship are from the same operational database;

p indicates the entity participating in the relationship;

e indicates the number of entities in operational database i;

q indicates the key attribute of the entity p;

f indicates the number of key attributes of the entity p;

{RA} is the optional set of relationship attributes and

$\{RA\} = \{\} \mid$
$\{A^r_{ij1}\} \mid$
$\{a_1, \ldots a_m\} \mid m \geq 2 \text{ and } a_y \in \{A^r_{ijk}, k=1,m\})$.

As an example, consider the case given below:

R_{11} $(\{A^k_{111}; A^k_{121}\}; \{A^r_{111}, A^r_{112}\})$
PRODUCES ({MANUF_ID; PROD_ID}; {UNITS_PROD, YR_PROD})

This notation shows that entity E_{11}, MANUFACTURER, and entity E_{12}, PRODUCT, are related, with the inclusion of their keys in the relation. Note that the two keys are first presented, separated by a semicolon because of the possibility that more than one attribute may form the key of an entity relation. The set of two keys is enclosed in brackets and followed by a semicolon to separate the list of key attributes from the list of relationship attributes. Finally, the optional set of relationship attributes is listed, with two attributes, UNITS_PROD and YR_PROD.

From our example, all real relationships are specified in Table V-2 below.

NATIONAL MANUFACTURING ORGANIZATION'S (E-R_1) DATABASE:
R_{11} ({A^k_{111}; A^k_{121}}; {A^r_{111}, A^r_{112}}) PRODUCES ({MANUF_ID; PROD_ID}; {UNITS_PROD, YR_PROD})
R_{12} ({A^k_{111}; A^k_{131}; A^k_{141}}; {A^r_{121}}) COLLECTS ({MANUF_ID; BY_PROD_ID; RECIP_ID}; {QTY_COLLECTED})
NATIONAL SHIPPING ORGANIZATION'S (E-R_2) DATABASE:
R_{21} ({A^k_{211}; A^k_{221}}; {A^r_{211}, A^r_{212}}) TRANSPORTS ({SHIP_CO_ID; SHIPMENT_NO}; {TRANSPORT_HAZARD})
ENVIRONMENTAL PROTECTION AGENCY (E-R_3) DATABASE:
R_{31} ({A^k_{311}; A^k_{321}}; {A^r_{311}, A^r_{312}}) DISPOSES({WDISPOSER_ID; WASTE_NO}; {QTY_DISPOSED, DISPOSAL_HAZARD})
R_{32} ({A^k_{311}; $A^k_{33}{}^1$}; {}) POSSESSES({WDISPOSER_ID; PERMIT_NO};{})

TABLE V-2: REAL RELATIONSHIPS IN EXAMPLE DATABASES

In the diagrams presented in this chapter, each E-R diagram which represents an operational database will be outlined using an rectangle. Entity-relationship diagrams for all operational databases from the example are presented in Figure V-2, using the notation outlined in this section.

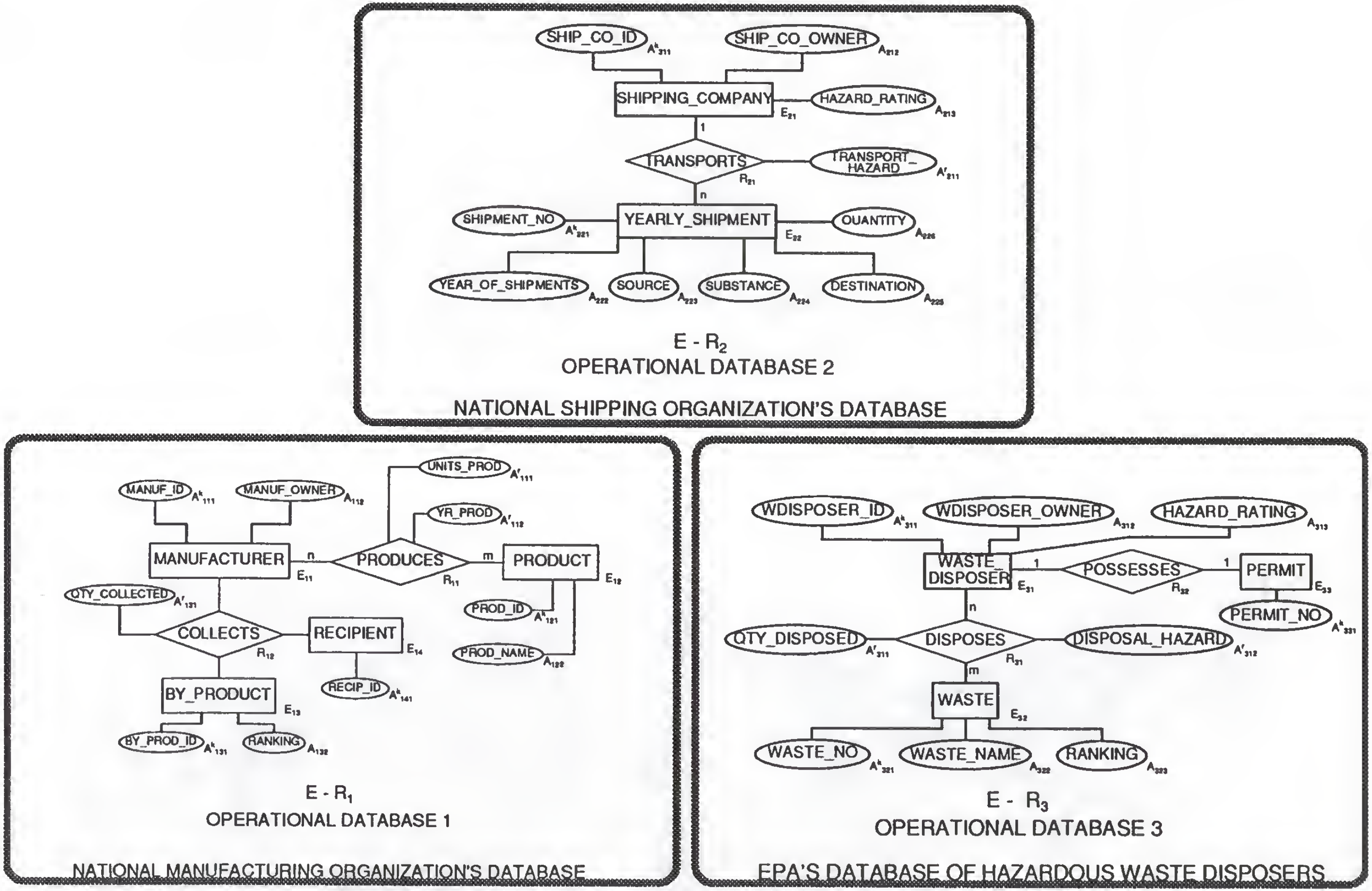


FIG. V-2: THREE EXAMPLE DATABASES

B.2 NOTATION FOR THE RESULTING VIRTUAL ENTITIES AND RELATIONSHIPS

The objective of the four-step procedure will be to identify virtual entities and relationships between them. Mappings from the operational databases to this virtual E-R will enable the DSS to retrieve information from the operational databases according to a common conceptual view of the data. The format for specifying information at the virtual level is given below.

A virtual entity VE_i is described by a subscript i which identifies the entity within the virtual E-R, a set of virtual attributes $\{VA_{ij}\}$, and a primary virtual key attribute (or set of attributes) $\{VA^k{}_{il}\}$.

A virtual relationship describes an association between two or more virtual entity sets. Each virtual relationship VR_i is described by a subscript i which identifies the relationship within the virtual E-R. The virtual relationship VR_i is described by the virtual keys of the entities being related and an optional set of relationship attributes.

VR_i $(\{vk_1, \ldots vk_n\} \mid n \geq 2;\ \{VRA\})$.

where $vk_x \in \{VA^k{}_{pq},\ p=1,e;\ q=1,f\}$;
- p indicates the virtual entity participating in the relationship;
- e indicates the number of entities in virtual E-R;
- q indicates the key attribute of the virtual entity p;
- f indicates the number of key attributes of virtual entity p;

{VRA} is the optional set of relationship attributes and

$\{VRA\} = \{\} \mid$
$\{VA^r{}_{i1}\} \mid$
$\{a_1, \ldots a_m\} \mid m \geq 2$ and $a_y \in \{VA^r{}_{ij},\ j=1,m\})$.

Example:
VR1 $(\{VA^k{}_{11};\ VA^k{}_{21};\ VA^k{}_{31}\};\{\})$

This notation indicates that the virtual entities VE_1, VE_2, VE_3 are related, with the inclusion of the key attributes for these virtual entities. As with real relationships, the keys are first presented, separated by a semicolon because of the possibility that more than one attribute may comprise the key of a virtual entity relation. The set of three keys is enclosed in brackets and followed by a semicolon to separate the list of key attributes from the list of relationship attributes. No attributes exist for this relationship.

Any relationship that holds between real entity sets which become virtual entities is a potential virtual relationship. The difficulty encountered in specifying these relationships is the determination of relationships that exist between virtual entities which do not directly correspond to real entities. For instance, these virtual entities may have been extracted from real entities or they may represent a combination of real entities. Virtual relationships will be specified by the DBA in the format given above; the process for developing these virtual entities and relationships will be given in the remainder of this chapter.

B.3. STEPS TOWARD SPECIFYING THE VIRTUAL E-R

Specification of the virtual E-R is a four-step process, as shown in Figure V-1. These four steps are outlined below and discussed in the following sections.

(1) identify mappings at the operational level
(2) identify relationships between real entities in different operational databases
(3) identify mappings from real to virtual entities
(4) identify relationships between virtual entities

B.3.1 (STEP 1) IDENTIFY MAPPINGS AT THE OPERATIONAL LEVEL

Mappings at the operational level can involve equivalences between three types of components of entity-relationship diagrams: (1) entities, (2) attributes of entities, and (3) attributes of relationships. In general, a mapping will consist of two parts: (1) an equivalence statement which tells the components which are semantically equivalent; and (2) a translation function, if necessary, which is used to translate values from one domain to another. The symbol $\equiv$ will be used to denote that two information items are equivalent; if translations are necessary, the function for translating values from one entity set into values for another entity set must be given as part of the mapping specified by the DBA.

Two components are considered equivalent whenever all of their values can be interpreted with the same meaning. Consider the BY_PRODUCTS and WASTE entity sets from the example databases. The same physical substance, hazardous waste, is represented in two different ways, but both representations have the same meaning. Because the two entity sets represent semantically equivalent objects, we say that the two entity sets are equivalent.

Both entity sets, BY_PRODUCT and WASTE, represent hazardous physical substances. However, each entity set describes different characteristics of a hazardous waste, and the key values for the two entity sets are from different domains. The domain for BY_PROD_ID allows alphanumeric values which contain waste codes published by the National Manufacturing Organization. In contrast, the domain for WASTE_NO allows numeric values which correspond to a waste code list published by the EPA. A conversion function is necessary to translate values from one domain to the other. This translation may be performed via a mathematical or other function; most likely, look-up tables will be necessary and must be provided by the DBA. Translations using a look-up table are illustrated in Table V-3 below.

BY_PROD_ID	OTHER DATA
BP001	
BP002	
BP003	
BP004	

TABLE V-3(a): BY_PRODUCT data

WASTE_NO	OTHER DATA
160	
170	
180	
190	

TABLE V-3(b): WASTE data

BY_PROD_ID	WASTE_NO
BP001	180
BP002	190
BP003	170
BP004	160

TABLE V-3(c): LOOK-UP TABLE FOR VALUES
BY_PROD_ID = f(WASTE_NO)
WASTE_NO = g(BY_PROD_ID)

TABLE V-3: LOOKUP TABLE FOR TRANSLATING KEY VALUES FOR ENTITY BY_PRODUCT INTO KEY VALUES FOR ENTITY WASTE

In this case, the look-up table is used to translate key values of one entity set into key values of another entity set; however, look-up tables may involve translation of non-key attribute values or relationship attribute values, depending upon the information items involved in the mapping.

The next section will discuss mappings which involve only entities and entity attributes. Mappings involving relationship attributes will then be addressed.

B.3.1.1 MAPPINGS BETWEEN REAL ENTITY SETS

Virtual entities will be formed by analyzing the entities in each of the operational databases and their relationships to other entities in the overall system; therefore, specifying mappings between entity sets is the preliminary step in identifying virtual entity sets. The DBA provides us with this information; this section outlines the format in which this information must be provided.

Recall the difference between a mapping and a relation. In this step, a mapping between real entity sets will establish an equivalence between two or more entity sets in different databases at the operational level. A real relationship, however, denotes an interaction between members of two or more entity sets at the operational level.

Like relationships, mappings can be specified as one-to-one, one-to-many and many-to-many. These different types of mappings are illustrated in Figure V-3 below.

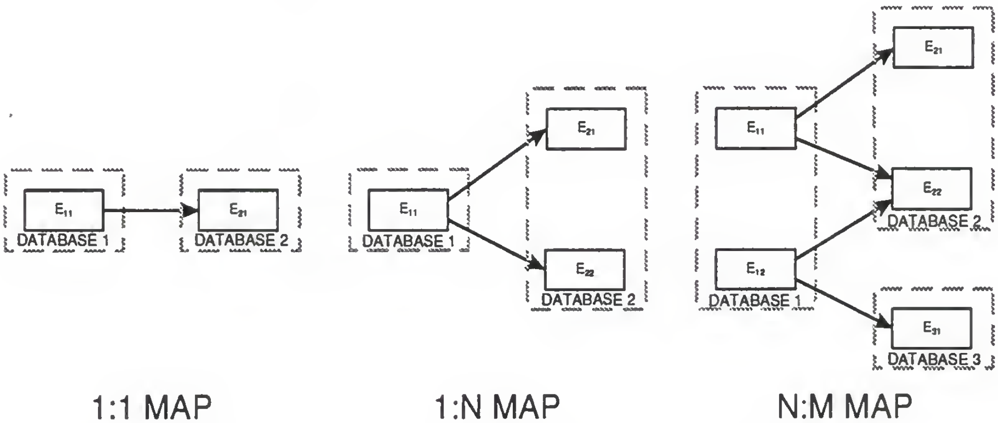


FIG. V-3: TYPES OF MAPPINGS AT THE OPERATIONAL LEVEL

A one-to-one mapping (1:1) is used to identify an equivalence between entity sets or parts of entity sets. A one-to-many (1:n) mapping identifies the case when information concerning one entity set is arranged in another database so that it describes n different entity sets. A many-to-many (n:m) mapping occurs whenever information concerning several n entity sets is arranged in another database so that it describes m different types of entities. Many-to-many mappings are rare and may be specified using a collection of one-to-many mappings rather than developing a separate notation for them.

A mapping between two entity sets indicates that the entity sets contain information about semantically equivalent objects. It must be possible to identify the tuples which describe a particular physical object. Therefore, each type of mapping will be specified in terms of key values for the tuples involved.

Mappings will be specified from an entity E_{ij} to entities in other operational databases. Two types of mappings may hold: a <u>whole entity mapping</u> will involve a complete entity on the left hand side of the equal sign, but a partial entity mapping will involve subsets of attributes of entity sets on both sides of the equal sign.

Whole entity mappings

Three different types of whole entity mappings are possible.

(1:1 MAPPINGS) (a) The entity set of interest E_{ij} is equivalent to an entity set which exists in another of the E-R_p's:

$$\mathbf{E_{ij} \equiv E_{pq}}.$$

Because the two entity sets are equivalent, their keys are equivalent; a translation function may be necessary to convert the key values from one entity set into the key values of the other entity set.

$$\therefore \quad A^k{}_{ijk} \equiv A^k{}_{pqr}.$$

$$\text{TRANSLATION FUNCTION: } A^k{}_{ijk} = t(A^k{}_{pqr})$$

Given the key value for an entity, the key values for equivalent tuples can be determined. This type of mapping will be referred to as a "one-to-one entity equivalence mapping."

(1:1 MAPPINGS) (b) The entity of interest E_{ij} is equivalent to a subset of attributes of an entity which exists in an E-R_p. In essence, the entity E_{ij} represents the same physical object that is formed as a projection of the entity E_{pq}:

$$\mathbf{E_{ij} \equiv \{A_{pqr},\ r=1,n\}}.$$

All attributes included in the projection are from the same real entity, E_{pq}, which we will refer to as the "projected entity." Assume that only one attribute is involved in the projection. Because a whole entity is equivalent to an attribute of another entity, the whole entity's key value must be equivalent to the attribute's value. A translation function may be necessary to convert the key values into the domain for the attribute values.

$$\therefore \quad A^k{}_{ijk} \equiv A_{pqr}.$$

$$\text{TRANSLATION FUNCTION: } A^k{}_{ijk} = t(A_{pqr}).$$

When tuples are selected which have the specified value for the attribute, more than one tuple may meet the selection criteria. Therefore, a function which takes a value of interest (the key for the whole entity) and produces a key value of the "projected entity" is not possible. However, all tuples which meet the selection criteria can be retrieved; each attribute value will be a function of the key of the "projected entity."

$$A_{pqr} = f(A^k{}_{pqs}).$$

Therefore, the relationship between the keys of the two entity sets is that each key of the entity set E_{ij} is equivalent to a function of the key of the projected entity set.

$$\therefore \quad A^k{}_{ijk} \equiv f(A^k{}_{pqs}).$$

Many tuples of the projected entity set can be associated with each tuple of the entity set E_{ij}. This type of mapping will be referred to as a "one-to-one entity projection mapping."

(1:n MAPPINGS) (c) The entity of interest consists of a concatenation of entities or attributes of entities from the different E-R's:

$$E_{ij} \equiv \prod_{y=1}^{n} X_y \quad \text{where } X_y = E_{ab} \mid A_{pqr} \text{ and } n \geq 2.$$

At least two operands, either entities or attributes of entities, must be concatenated in a one-to-many entity mapping. The key of the resulting entity will be a function of the keys of the concatenated entities as well as the concatenated attributes. In addition, some or all of the keys may be translated.

$$A^k{}_{ijk} \equiv f(\text{keys of } X_y,\ y=1,n).$$
TRANSLATION FUNCTION: $A^k{}_{ijk} = t(\text{keys of } X_y, y=1,n)$.

This type of mapping will be referred to as a "one-to-many entity mapping."

Partial entity mappings

A fourth type of mapping does not involve a whole entity on the left hand side of the equal sign, but rather selected attributes of entities on both sides. This mapping occurs whenever certain attributes of one entity represent a "sub-entity" that is equivalent to a set of attributes of another entity. All attributes on one side of the equal sign must be a projection from the same real entity. That is, a combination of one attribute of an entity with an attribute from another entity to represent a physical object is not allowed. Hence, this mapping is a one-to-one mapping between entity sets, because only one entity set is represented on each side of the equal sign.

$$\{A_{ijk},\ k=1,m\} \equiv \{A_{pqr},\ r=1,n\}.$$

Assume that only one attribute is involved on each side of the mapping; a translation may be required to convert the values of one attribute into values in the domain of the other attribute.

TRANSLATION FUNCTION: $A_{ijk} = t(A_{pqr})$.

Again, any non-key attribute's value will be a function of the key attribute's value.

$$\text{for all } A_{ijk},\ k=1,m\text{:}\quad A_{ijk} = f(A^{k}_{ijl}) \quad \text{AND}$$

$$\text{for all } A_{pqr},\ r=1,m\text{:}\quad A_{pqr} = g(A^{k}_{pqs}).$$

$$\therefore f(A^{k}_{ijl}) \equiv g(A^{k}_{pqs}).$$

In this case, there is no direct relationship between the key values of the two entities involved in the mapping. Tuples must be selected from each of the two entity sets, and then combined to obtain information about a single physical object.

Each of these types of mappings will be discussed in a separate section below. Because the purpose of specifying real entity mappings is to identify how the semantically equivalent real entities should be represented at the virtual level, each section will show how the information gained from identifying these mappings will be used.

B.3.1.1.1 EXAMPLES AND DISCUSSION OF ONE-TO-ONE ENTITY EQUIVALENCE MAPPINGS

In the example, we have two cases of one-to-one entity equivalence mappings:

BY_PRODUCT	≡	WASTE	$E_{13} \equiv E_{32}$
RECIPIENT	≡	SHIPPING COMPANY	$E_{14} \equiv E_{21}$

These mappings are illustrated in Figure V-4 on the next page.

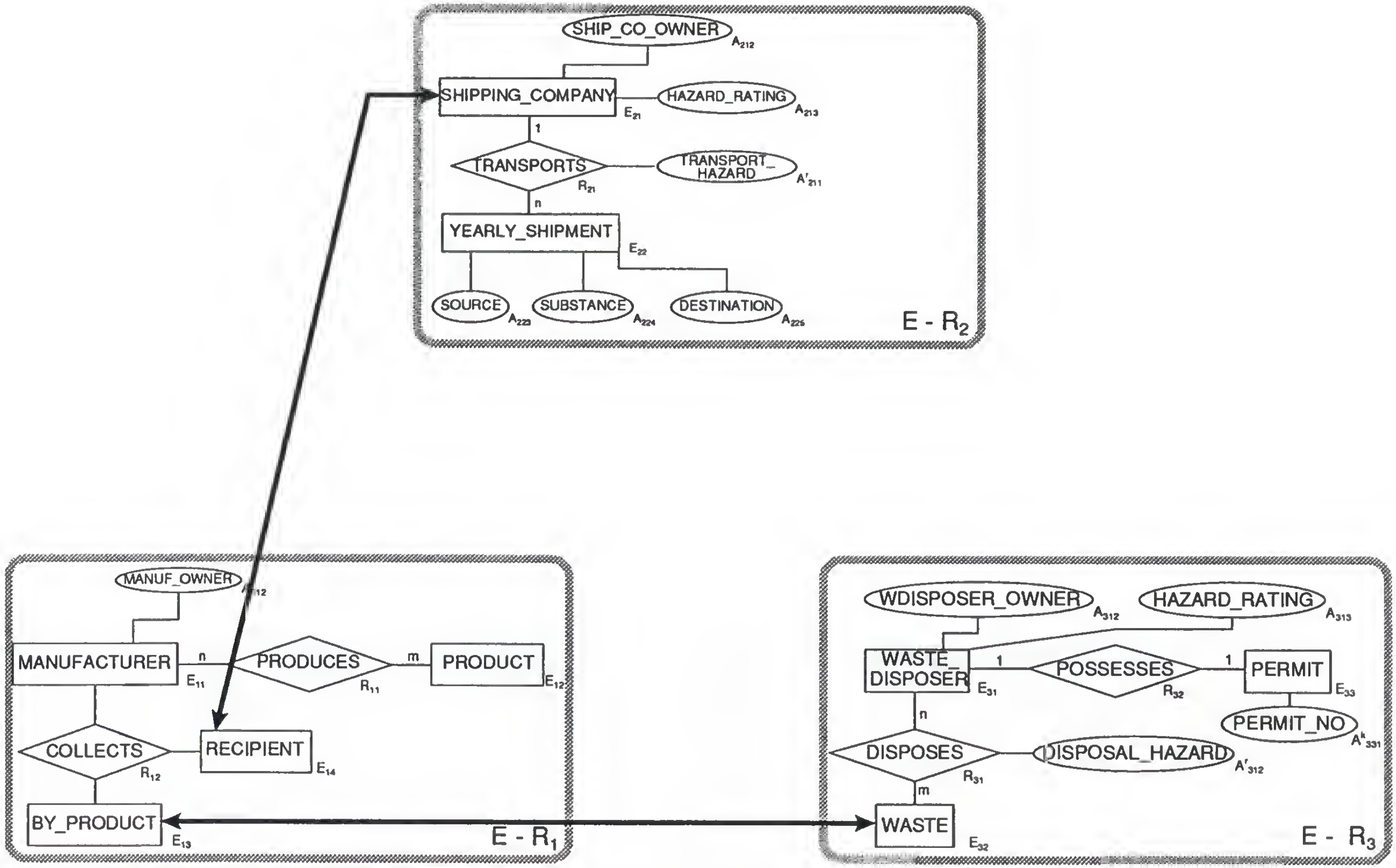


FIG. V-4: ONE-TO-ONE ENTITY EQUIVALENCE MAPPINGS

A one-to-one equivalence mapping does not imply that the key values for each entity will be equal; a translation between key values may be necessary. This translation may be performed via a mathematical function; most likely, a look-up table will be provided by the DBA. For implementation, the function which performs the translation must be specified. In this example, the look-up table shown in Figure V-2 will be used to translate values. Once the look-up function has been applied to one key value, all equivalent tuples are identified.

The information conveyed by this mapping will be used in developing the virtual representation of the data contained in these entities. Because the two entities are semantically equivalent, it is possible to represent them using only one entity in the virtual E-R. However, it may be necessary to examine the attributes of each entity relation in the operational databases, as it is likely that one of the databases contains information about the physical object that the other does not. When forming the virtual entity, each of the attributes from each real entity set should be contained in the single representation of the object at the virtual level.

In implementing the one-to-one equivalence mapping for the WASTE from the EPA database and the BY_PRODUCT from the MANUFACTURER'S database, consider the entity attributes. Semantically, these two entities represent the same type of physical substance. Note that each of these entities has a RANKING attribute. In the WASTE relation, RANKING represents the toxicity score for the waste, but in the BY PRODUCT entity, RANKING represents an inflammability index. Therefore, these two attributes represent different characteristics of the same physical substance, even though they have the same name. Each of the RANKING attributes should be maintained, and at least one of them should be renamed at the virtual level. Therefore, mappings between attributes will be necessary too; this topic will be addressed in the section of this paper which describes the formation of virtual entities.

B.3.1.1.2 EXAMPLES AND DISCUSSION OF ONE-TO-ONE ENTITY PROJECTION MAPPINGS

These mappings illustrate the following reality of working with decision support systems: the same piece of information may be of differing importance in different application areas. In particular, information that is represented as an entity in one database may be represented as attributes of another entity in a different database.

In the example databases, there are four examples of one-to-one projection mappings:

BY_PRODUCT	≡	SUBSTANCE	$E_{13} \equiv A_{224}$
WASTE	≡	SUBSTANCE	$E_{32} \equiv A_{224}$
MANUFACTURER	≡	SOURCE	$E_{11} \equiv A_{223}$
WASTE_DISPOSER	≡	DESTINATION	$E_{31} \equiv A_{225}$.

These mappings are illustrated in Figure V-5 on the next page.

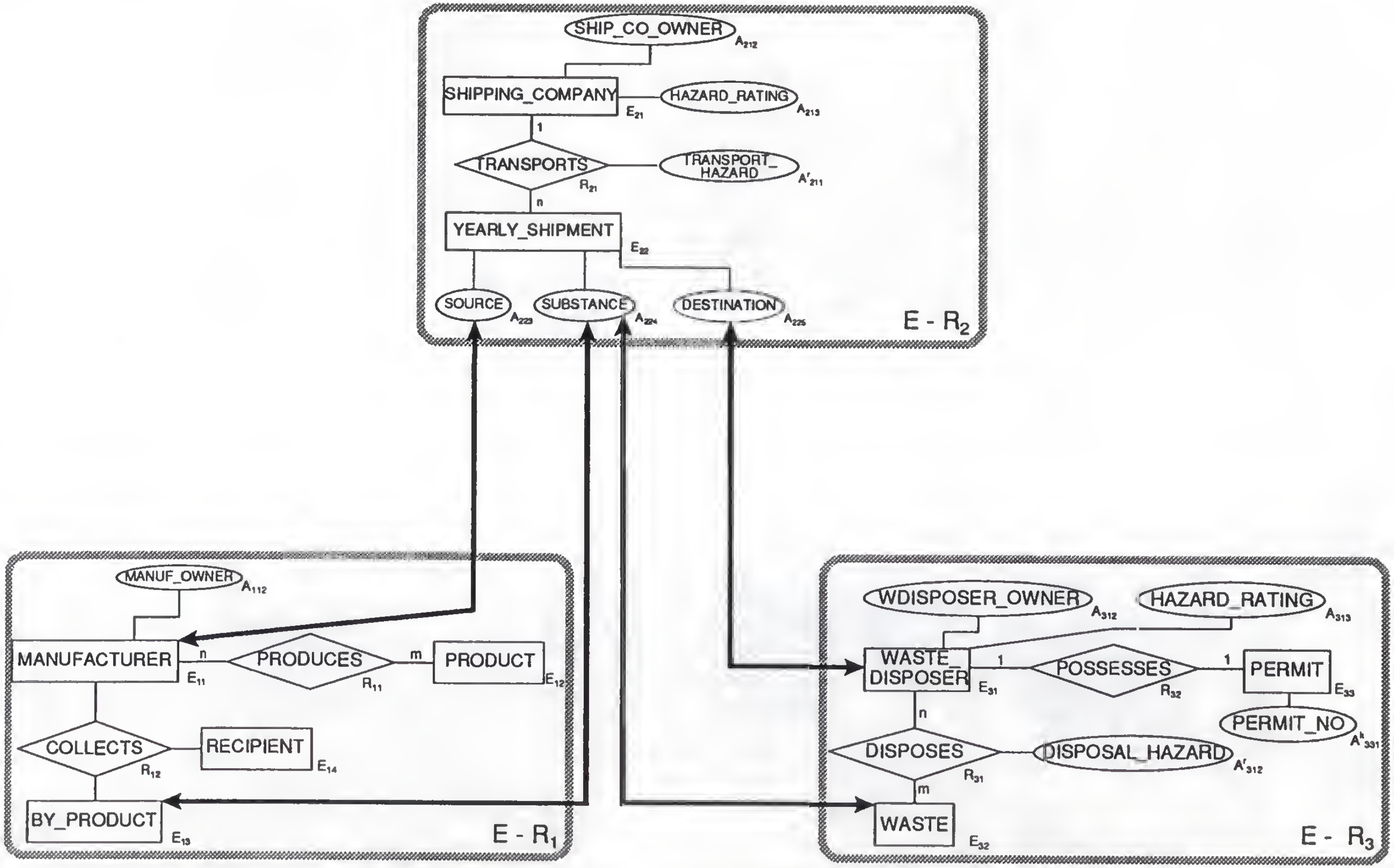


FIG. V-5: ONE-TO-ONE ENTITY PROJECTION MAPPINGS

For example, WASTE is viewed as an important unit of information to the EPA database designer and is represented as an entity. The designer of the SHIPPING COMPANY'S database viewed SUBSTANCE as an attribute of a more important information entity, a SHIPMENT. For the problem situation confronted by the DBA, the physical WASTE entity being described may be of critical importance and require representation as a separate entity, or it may not. It is important, however, that the semantic equivalence be identified so that the DBA can consider how to represent the information at the virtual level.

A one-to-one projection mapping indicates that a subset of attributes of one entity represents all of the information maintained about a second entity. A new "projected entity" will be created that contains only the attributes of the first entity that have been specified in the mapping. In this example, the "projected entity" consists of the single attribute SUBSTANCE.

The objective is to determine whether a single virtual entity can be used to represent both sides of the mapping. This determination will depend upon whether the entity on which the projection is performed is involved in other mappings and whether it is of interest to the problem domain. Usually, the real entity would be chosen to be mapped to the virtual level, assuming that the "projected entity" would be covered. However, the DBA must carefully consider whether all of the attributes of the real entity apply to the new "projected entity".

If the real entity is chosen to represent itself in addition to the new "projected entity" at the virtual level, the impact on the first real entity which originally contained the attributes of the new "projected entity" must be considered. If this first real entity is also mapped to the virtual level, should it still contain those attributes which became the projected entity? In our example, if the YEARLY_SHIPMENT and the WASTE entities are mapped to the virtual level, should the YEARLY_SHIPMENT virtual entity retain the SUBSTANCE attribute? The alternative is to show the fact that a YEARLY_SHIPMENT contains a particular SUBSTANCE (WASTE) via a relationship. The DBA must carefully consider these points in deciding what is mapped to the virtual level. The important point at this step is to identify all equivalences so that they can be considered.

B.3.1.1.3 EXAMPLES AND DISCUSSION OF ONE-TO-MANY ENTITY MAPPINGS

A one-to-many mapping indicates that information describing a single entity in one operational database has been split into several entities in another operational database. Three possible virtual representations exist in this case: (1) choose the single entity to represent itself as well as the N other entities; (2) choose m entities, where $m \leq N$, to exist at the virtual level and to represent the N+1 entities; or (3) map all N+1 entities to the virtual level. The choice of representation will largely depend upon the amount of information needed at the virtual level, the contribution of each entity to the information needed, the relationships of each entity with other real entities, and the chosen mapping of those related entities to the virtual level.

Our database example does not have a real entity which represents a concatentation of other entities. A many-to-one mapping will be illustrated in the discussion of mappings from real to virtual entities.

B.3.1.1.4 EXAMPLES AND DISCUSSION OF ONE-TO-ONE ATTRIBUTE MAPPINGS

In this mapping, the information represented has not been considered sufficiently important to either application area to be represented at the entity level. However, the attributes semantically represent the same physical object, which will be referred to as a "subentity". The subentity may be considered critical to the problem situation and should be identified so that it may be considered as a potential virtual entity.

The example contains four cases of one-to-one attribute mappings.

MAN_OWNER ≡ SHIP_CO_OWNER	$A_{112} \equiv A_{212}$
MAN_OWNER ≡ WDISPOSER_OWNER	$A_{112} \equiv A_{312}$
SHIP_CO_OWNER ≡ WDISPOSER_OWNER	$A_{212} \equiv A_{312}$
HAZARD_RATING ≡ HAZARD_RATING	$A_{213} \equiv A_{313}$

These mappings are illustrated in Figure V-6 on the next page.

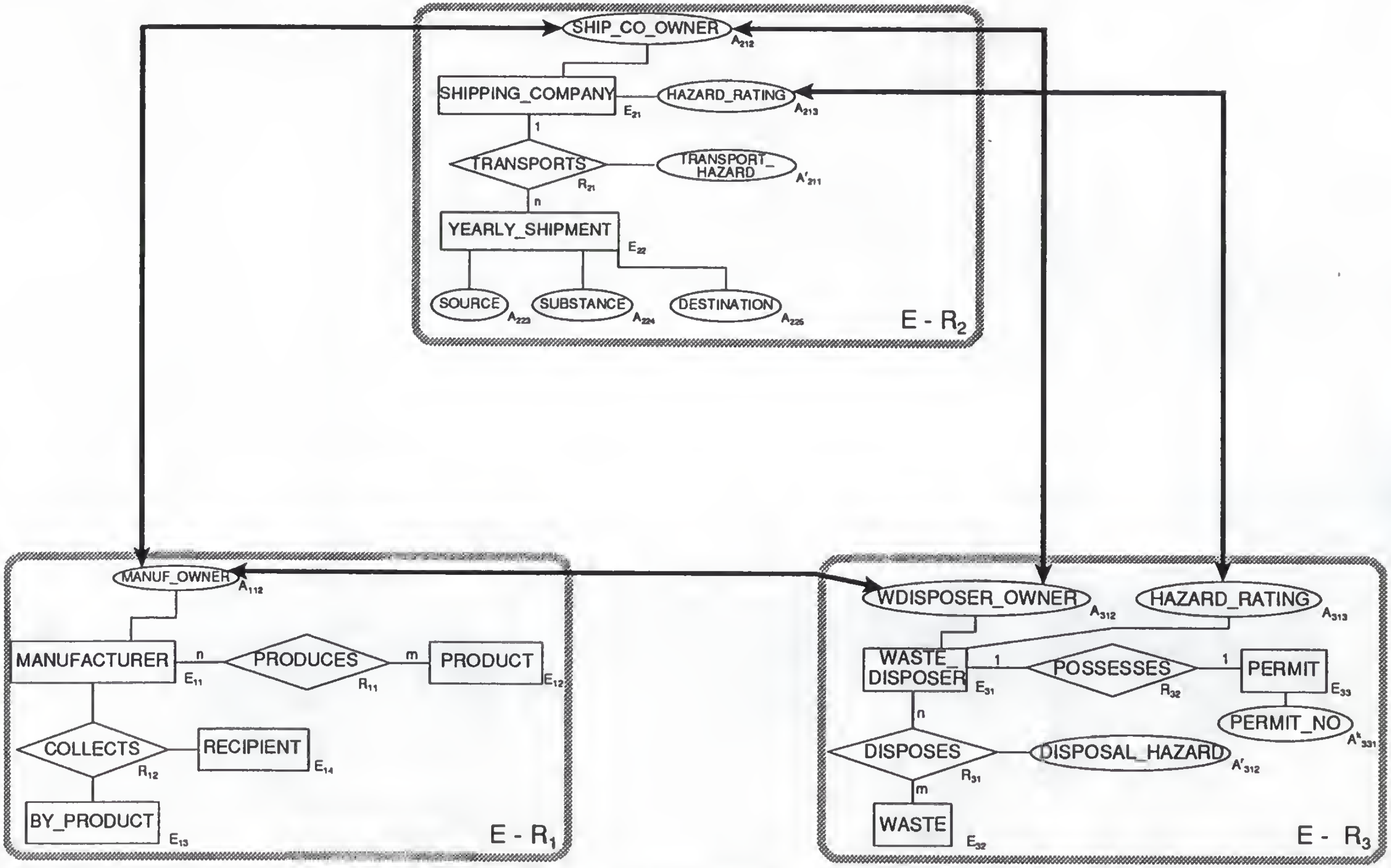


FIG. V-6: ONE-TO-ONE ENTITY ATTRIBUTE MAPPINGS

An example of a one-to-one attribute mapping is the mapping between the various attributes which describe the owners of the business enterprises represented in the three databases. In the MANUFACTURER entity, the MAN_OWNER attribute represents the person or business which owns the manufacturing company; the SHIP_CO_OWNER attribute of SHIPPING COMPANY and the WDISPOSER_OWNER attribute of WASTE DISPOSER serve similar purposes. Although the database designers did not choose to represent the owners as separate entities, the owner of a business operation involved in handling hazardous waste is of critical importance to our example problem situation. If a single owner is involved in the handling of hazardous waste from generation to disposal, this information is important for purposes of identifying responsible parties. By describing this equivalence, the DBA can consider whether these attributes should be represented as an entity at the virtual level.

B.3.1.2 EXAMPLES AND DISCUSSION OF MAPPINGS INVOLVING RELATIONSHIP ATTRIBUTES

Relationships may have unique attributes of their own. Just as attributes of entities can be mapped to either whole entities or attributes of other entities, so can attributes of relationships.

The three possible types of mappings involving relationship attributes are given below:

One-to-one Relationship Attribute Mapping

In this mapping, an attribute of one relationship represents the same physical object as an attribute of another relationship. All attributes on one side of the equivalence sign must be from the same real relationship.

$$\{\mathbf{A^r{}_{ijk},\ k=1,n}\} \equiv \{\mathbf{A^r{}_{xyz},\ z=1,n}\}.$$

Assume that only one relationship attribute is involved in the mapping and that relationships R_{ij} and R_{xy} involve two entities each.

$$\text{TRANSLATION FUNCTION:}\quad A^r{}_{ijk} = t(A^r{}_{xyz})$$

Each relationship attribute is a function of the keys of the entities involved in the relationship.

$$A^r{}_{ijk} = f(A^k{}_{pqr},\ A^k{}_{abc}) \text{ and}$$

$$A^r{}_{xyz} = g(A^k{}_{def},\ A^k{}_{ghi}).$$

$$\therefore\ f(A^k{}_{pqr},\ A^k{}_{abc}) \equiv g(A^k{}_{def},\ A^k{}_{ghi}).$$

There is no direct equivalence between the keys of the entities involved in the relationships. In order to determine tuples which meet the requirements for this mapping, relationship tuples must be retrieved from each relationship set which meet the selection criteria for the relationship attribute.

In the example databases, an equivalence occurs between the TRANSPORT_HAZARD attribute of the TRANSPORTS relationship and the DISPOSAL_HAZARD attribute of the DISPOSES relationship. For each relationship, a hazard rating is assigned according to the characteristics of the waste handler in addition to the toxicity of the substance being handled.

TRANSPORT_HAZARD ≡ DISPOSAL_HAZARD $\qquad A^r{}_{211} \equiv A^r{}_{311}.$

This mapping is illustrated in Figure V-7 on the next page.

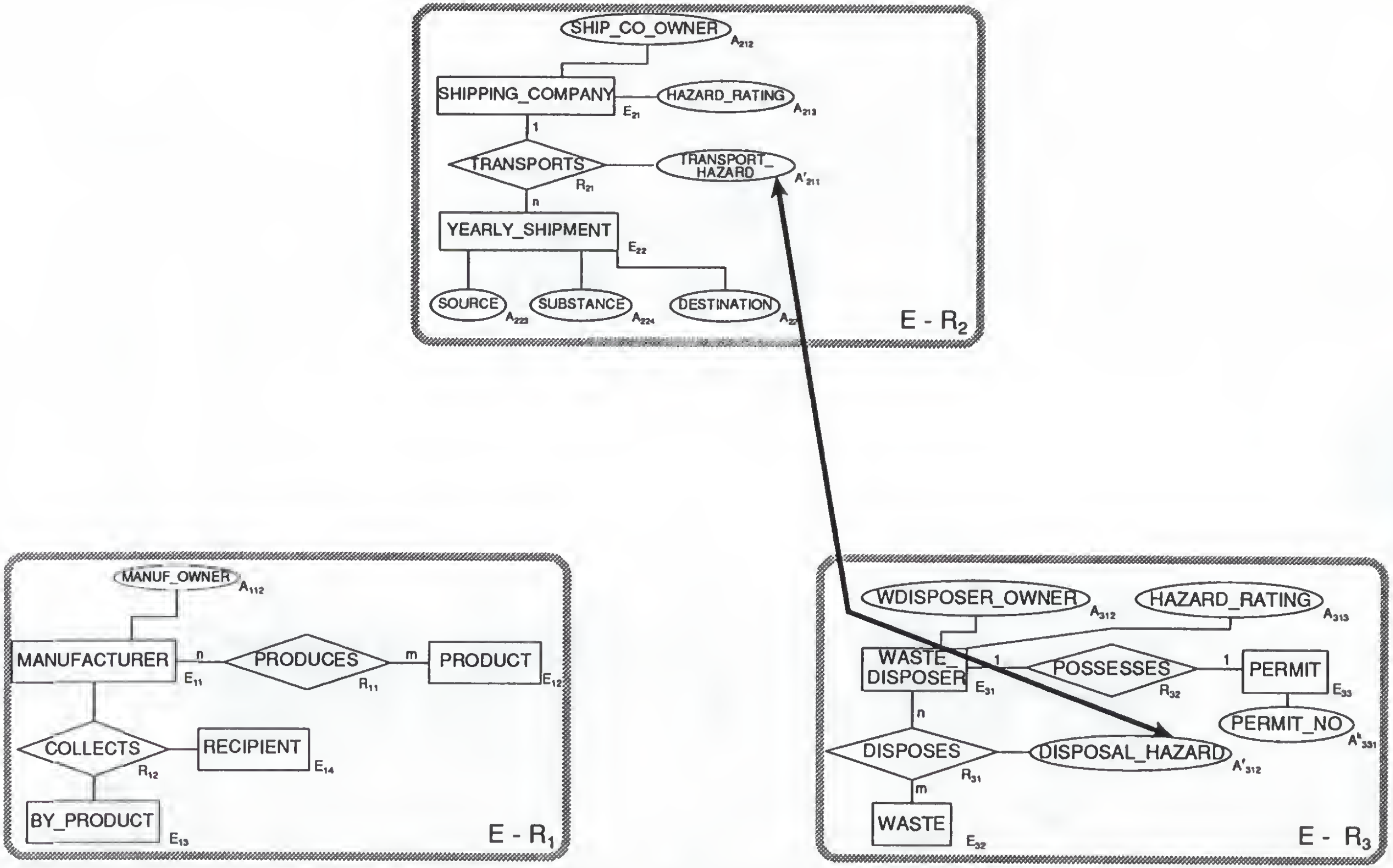


FIG. V-7: RELATIONSHIP ATTRIBUTE TO RELATIONSHIP ATTRIBUTE MAPPING

One-to-one Relationship Attribute to Entity Mapping

In this type of mapping, an attribute of a relationship represents the same physical object as an entity. All attributes on the left side of the equivalence sign must be from the same real relationship, R_{ij}.

$$\{A^r_{ijk},\ k=1,n\} \equiv E_{xy}.$$

Assume that only one relationship attribute is involved in the mapping and that relationship R_{ij} involves only two entities. Because the entity is equivalent to an attribute, the entity's key value must be equivalent to the attribute's value; a translation function may be necessary to establish this equality.

$$\therefore A^r_{ijk} \equiv A^k_{xyz}.$$

$$\text{TRANSLATION FUNCTION:} \quad A^r_{ijk} = t(A^k_{xyz}).$$

A relationship attribute is a function of the keys of the entities involved in the relationship.

$$A^r_{ijk} = f(A^k_{pqr},\ A^k_{abc}).$$

Therefore, the relationship attribute is equivalent to a function of the keys of the entities involved in the relationship.

$$\therefore f(A^k_{pqr},\ A^k_{abc}) \equiv A^k_{xyz}.$$

In order to form tuples which meet the requirement for this mapping, relationship tuples must be retrieved which meet the selection criteria for the relationship attribute. No direct equivalence exists between the key of the entities involved in the relationship and the key of the entity involved in the mapping.

The example databases do not have an example of a relationship-attribute-to-entity mapping.

One-to-one Relationship Attribute to Entity Attribute Mapping

In this type of mapping, an attribute of a relationship represents the same physical object as an attribute of an entity. All attributes on one side of the equivalence sign must be from the same real entity or relationship.

$$\{A^r ijk,\ k=1,m\} \equiv \{Apqr,\ r=1,m\}.$$

Assume that only one relationship attribute is involved in the mapping, only one entity attribute is involved in the mapping, and relationship R_{ij} involves only two entities. Values for the relationship attribute may need translation in order to meet the domain requirements for the entity attribute.

TRANSLATION FUNCTION: $A^r{}_{ijk} = t(A_{pqr})$.

A non-key entity attribute's value is a function of the key entity attribute's value.

$$\therefore A^r{}_{ijk} \equiv g(A^k{}_{pqs}).$$

A relationship attribute is a function of the keys of the entities involved in the relationship.

$$A^r{}_{ijk} = f(A^k{}_{xyz}, A^k{}_{abc}).$$

$$\therefore f(A^k{}_{xyz}, A^k{}_{abc}) \equiv g(A^k{}_{pqs}).$$

There is no direct equivalence between the keys of the entities involved in the relationship and the key of the projected entity. In order to identify tuples which meet the criteria for this mapping, entities with the specified entity attribute must be retrieved and matched against relationship tuples with the specified relationship attribute.

The example databases do not have an example of a relationship-attribute-to-entity-attribute mapping.

The purpose of these mappings is to show how much information is available in the databases about the physical objects being described. Therefore, the DBA can determine whether enough information can be contributed from an object to warrant consideration as a virtual entity.

Another purpose for identifying mappings involving relationship attributes is described below. If a physical object is described as both a relationship attribute and a real entity, then new relationships may be introduced across database boundaries. Several types of new associations, or relationships, are the topic of the next chapter.

All of the semantic equivalences between different components of the operational databases have now been identified. All of the mappings established from the analysis of information in the operational databases are illustrated in Figure V-8 below.

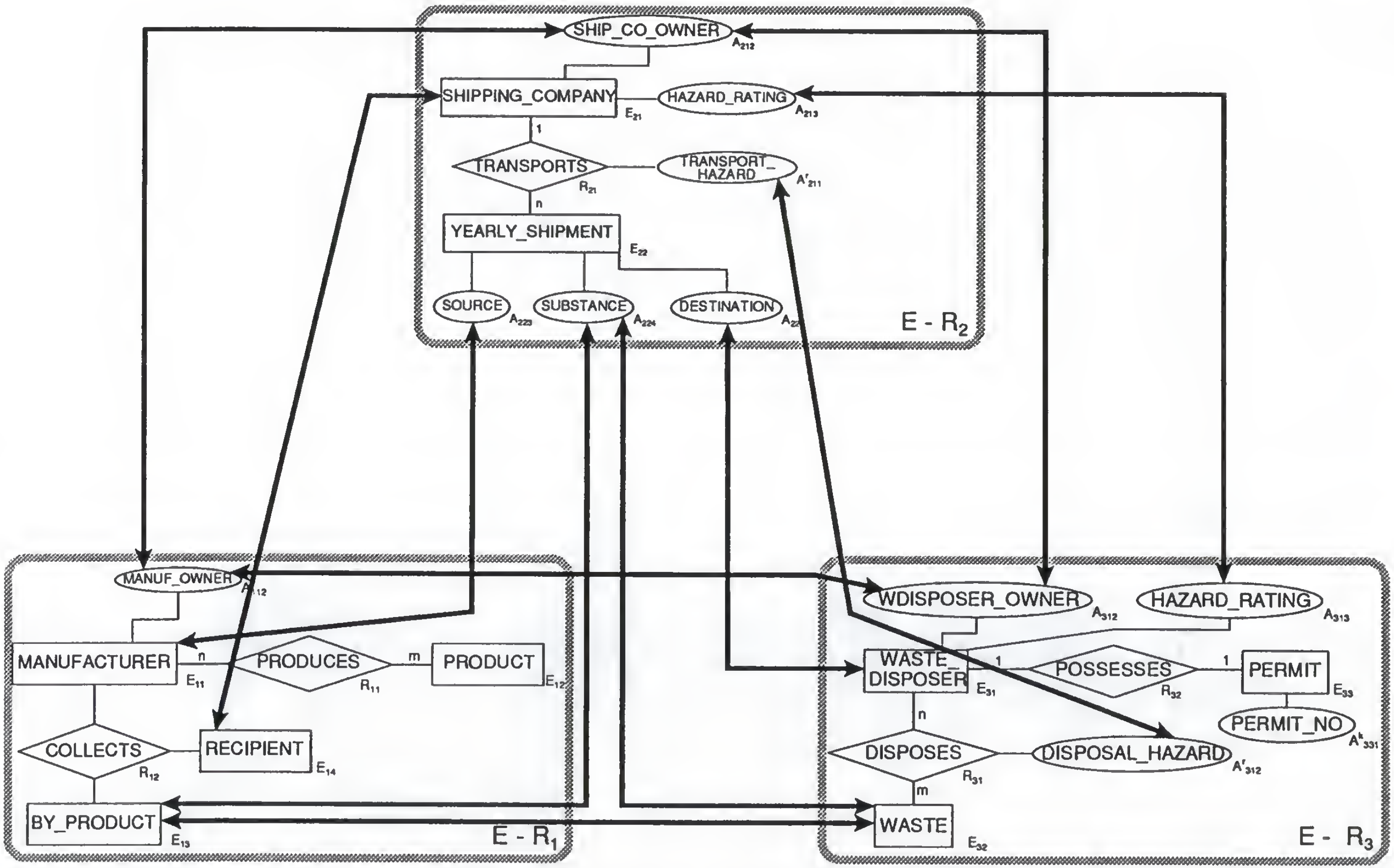


FIG. V-8: ALL MAPPINGS AT OPERATIONAL LEVEL

These mappings provide us with information that will enable us to determine the appropriate virtual representation of the data in the operational databases. Using these mappings, we will identify relationships between entities in different operational databases in the next section of this chapter.

B.3.2 (STEP 2) IDENTIFY RELATIONSHIPS BETWEEN REAL ENTITIES IN DIFFERENT OPERATIONAL DATABASES

In Step 1, the DBA is to identify semantic equivalences between components of different database systems. If an entity set in one database is equivalent to a second entity set in another database, the relationships in which the first entity set is involved may also apply to the second equivalent entity set. Also, a mapping between two previously unrelated entities may introduce a relationship. The key feature of these relationships is that they relate entities across operational database boundaries.

These relationships are used to extract "hidden" semantic information based upon the DBA's global view of all of the databases and the problem domain. This information is not currently represented within any operational database; if it is to be represented at all in the DSS, the DBA must specify it. He can then assess whether the information is of sufficient value to be added as a virtual relation. Of course, whether these relationships can be directly translated into virtual relations will depend upon how the related real entities are translated into virtual entities. If the two real entities are translated using a one-to-one equivalence mapping, then these real relationships are automatically suited to become a virtual relationship.

B.3.2.1 NOTATION FOR SPECIFYING RELATIONSHIPS BETWEEN REAL ENTITIES IN DIFFERENT OPERATIONAL DATABASES

A <u>relationship between real entities in different operational databases</u> RR_i is described by a subscript i which identifies the relationship. Note that there is no subscript to describe the E-R in which the relationship resides, since the relationship crosses database boundaries. Entities must, however, continue to be identified using the double-subscript notation. In accordance with Chen's model, we will restrict relationships to those between two or more whole entities. That is, the model will not allow specification of a relationship that involves only a subset of attributes of entities.

The same type of notation will be used as that used for real relationships, except that relationship attributes will have a second superscript to distinguish them from attributes of the given real relationships. This distinction may be important because of mappings specified involving relationship attributes, which were described above.

Therefore, the relation RR_i will be described by the real keys for the entities participating in the relationship, followed by an optional list of relationship attributes. The notation for specifying these relationships is given below.

RR_i ({k_1, ...k_n} | n≥2; {RRA}).

where k_x ε {$A^k{}_{pqr}$, p=1,d; q=1,e; r=1,f};
p indicates the operational database, or E-R;
d indicates the number of operational databases;
q indicates the entity participating in the relationship;
e indicates the number of entities in operational database p;
r indicates the key attribute of the entity q;
f indicates the number of key attributes of entity q;
{RRA} is the optional set of relationship attributes and
{RRA} = {} |
{$A^{rr}{}_{i1}$} |
{a_1, ...a_m} | m≥2 and a_y ε {$A^{rr}{}_{ij}$, j=1,m}).

PROCEDURE: $RR_i \equiv R_{ij}$ OR OTHER SPECIFICATION.

A <u>transferred relationship</u> results whenever two entities are considered to be equivalent, and because of the equivalence, a relationship involving one of the entities will also hold for the second entity. The transferred relationship will assume the same relationship name as the original relationship, with a _T appended. Relationship attribute names will also have _T appended. If a relationship is transferred more than once, each additional transferred relationship will be sequentially numbered, with the number appended after the _T.

The association between relationship attributes and entity keys is stated below:

$$A^{rr}{}_{ij} = f(\{k_1, \ldots k_n\})$$

Because these relationships are between entities in different operational databases, we must also specify how tuples for the relationships will be derived. If an entity set in one database is equivalent to a second entity set in another database, the relationships in which the first entity set is involved may be transferred to the second equivalent entity set. Therefore, the RR_i may be equivalent to a real relationship R_{jk}. Because an entity equivalence mapping may require the translation of key values in order to establish the equivalence, the formation of a relationship between entities may also require the translation of key values for entity sets.

A new relationship between entity sets may also be introduced by a mapping between two previously unrelated entities. In this case, a procedure must be specified that determines which entities are involved in this new relationship.

B.3.2.2 HOW INFORMATION FROM THE PREVIOUS STEP WILL BE USED

In order to identify the entities which are involved in these relationships, information from Step 1 will be used. As we noted above, mappings at the operational level are used to identify the semantic equivalences between different components of the operational E-R diagrams. If an entity set is mapped to a second entity set in another database, then the possibility exists that any other entities related to a member of the first entity set are also related to the members of the second equivalent entity set. We describe the new relationship as a "transferred relationship", as illustrated in Figure V-9 below.

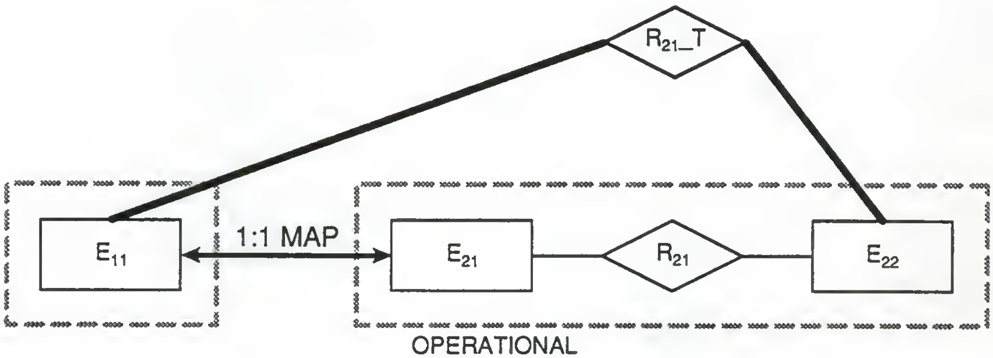


FIG. V-9: TRANSFERRED RELATIONSHIP AS A RESULT OF A ONE-TO-ONE ENTITY EQUIVALENCE MAPPING

The mappings specified previously will enable us to identify relationships which exist across database boundaries. Although a relationship involves two or more entity sets, any type of mapping can potentially introduce a new relationship. For instance, mappings which involve relationship attributes may also introduce new relationships. Because a relationship attribute's value is determined by a combination of values of entity key attributes, its equivalence to a component of another database may introduce new relationships between the involved entities. Therefore, all types of mappings from Step 1 will be examined. All mappings from the example databases were illustrated in Figure V-8, and are presented below in Table V-4.

One-to-one entity equivalence mappings:	
RECIPIENT ≡ SHIPPING COMPANY	$E_{14} \equiv E_{21}$
BY_PRODUCT ≡ WASTE	$E_{13} \equiv E_{32}$
One-to-one projection mappings:	
BY_PRODUCT ≡ SUBSTANCE	$E_{13} \equiv A_{224}$
WASTE ≡ SUBSTANCE	$E_{32} \equiv A_{224}$
MANUFACTURER ≡ SOURCE	$E_{11} \equiv A_{223}$
WASTE_DISPOSER ≡ DESTINATION	$E_{31} \equiv A_{225}$
One-to-many entity mappings: none	
One-to-one attribute mappings:	
MAN_OWNER ≡ SHIP_CO_OWNER	$A_{112} \equiv A_{212}$
MAN_OWNER ≡ WDISPOSER_OWNER	$A_{112} \equiv A_{312}$
SHIP_CO_OWNER ≡ WDISPOSER_OWNER	$A_{212} \equiv A_{312}$
HAZARD_RATING ≡ HAZARD_RATING	$A_{213} \equiv A_{313}$
Relationship-attribute-to-relationship-attribute mappings:	
TRANSPORT_HAZARD ≡ DISPOSAL_HAZARD	$A^{r}_{211} \equiv A^{r}_{311}$
Relationship-attribute-to-entity mappings: none	
Relationship-attribute-to-entity-attribute mappings: none	

TABLE V-4: MAPPINGS FROM STEP 1

B.3.1.3 EXAMPLES AND DISCUSSION OF RELATIONSHIPS BETWEEN ENTITIES IN DIFFERENT OPERATIONAL DATABASES

For the example databases, each eligible mapping from Step 1 is examined to determine whether additional relationships are introduced as a result of the mapping.

One-to-one entity equivalence mappings:

Consider the one-to-one entity equivalence mapping

RECIPIENT ≡ SHIPPING COMPANY $E_{14} \equiv E_{21}$.

Because this mapping is a one-to-one equivalence mapping, the two entity sets represent the same physical object. In this case, the object is a company which picks up hazardous wastes from manufacturing sites. RECIPIENT entities are involved in the COLLECTS relationship with entities from WASTE and MANUFACTURER; RECIPIENT'S equivalence with SHIPPING COMPANY implies that the SHIPPING COMPANY is also related to the WASTE and MANUFACTURER entities. This new relationship is a "transferred relationship." The formal notation for the relationship is given below.

RR_1 ({$A^k{}_{111}$; $A^k{}_{211}$; $A^k{}_{131}$}; {$A^{rr}{}_{11}$})
COLLECTS_T({MANUF_ID; SHIP_CO_ID; BY_PROD_ID}; {QTY_COLLECTED_T})
PROCEDURE: $RR_1 \equiv R_{12}$

This notation indicates that real entities E_{11}, MANUFACTURER, E_{21}, SHIPPING COMPANY, and E_{13}, BY_PRODUCT, are related. The relationship between these three entities also generates its own attribute, $A^{rr}{}_{11}$, QTY_COLLECTED_T. This relationship involving three entities is transferred from the real relationship COLLECTS, R_{12}, because of the one-to-one equivalence mapping between RECIPIENT and SHIPPING COMPANY. The relationship name, COLLECTS_T, is derived from the name of the original relationship, with a T appended to denote that the relationship has been transferred. Relationship attribute names also have a T appended.

Because the values for RECIP_ID and SHIP_CO_ID have the same domain, tuples for this relationship are the same as the tuples for the COLLECTS relationship from Chapter IV and will not be repeated here.

The equivalence of SHIPPING COMPANY and RECIPIENT also implies that the TRANSPORTS relationship can be transferred to the RECIPIENT entity so that RECIPIENT is related to the YEARLY SHIPMENTS entity. This relationship is specified below.

RR_2 ({$A^k{}_{141}$; $A^k{}_{221}$}; {$A^{rr}{}_{21}$})
TRANSPORTS_T ({RECIP_ID; SHIPMENT_NO}; {TRANSPORT_HAZARD_T})
PROCEDURE: $RR_2 \equiv R_{21}$

Because the values for RECIP_ID and SHIP_CO_ID have the same domain, tuples for this relationship are the same as the tuples for the TRANSPORTS relationship from Chapter IV and will not be repeated here.

The second equivalence mapping from Step 1 is the mapping from BY_PRODUCT to WASTE,

BY_PRODUCT ≡ WASTE $\qquad E_{13} \equiv E_{32}$.

This mapping results in two transferred relationships:

RR_3 ({$A^k{}_{111}$; $A^k{}_{211}$; $A^k{}_{321}$}; {$A^{rr}{}_{31}$})
COLLECTS_T2 ({MANUF_ID; SHIP_CO_ID; WASTE_NO};{QTY_COLLECTED_T2})
$RR_3 \equiv R_{12}$

Tuples for this relationship are shown below. Note that these tuples are the same as the tuples for the COLLECTS relationship from Chapter

IV, except that the values for WASTE_NO have been translated into values for BY_PROD_ID according to the look-up table in Table V-3.

MANUF_ID	WASTE_NO	RECIP_ID	QTY_COLLECTED
M050	190	SC05	114
M100	160	SC20	10
M250	160	SC20	25
M200	180	SC10	50

TABLE V-5: COLLECTS_T2 relationship data

Because this is the second equivalence mapping which has involved the COLLECTS relationship, a 2 has been appended to the relationship name, COLLECTS_T2, and to its attribute name QTY_COLLECTED_T2.

The second transferred relationship as a result of this mapping is shown below.

RR_4 $(\{A^k{}_{131}; A^k{}_{311}\}; \{A^{rr}{}_{41}, A^{rr}{}_{42}\})$
DISPOSES_T ({BY_PROD_ID; WDISPOSER_ID}; {QTY_DISPOSED_T, DISPOSAL_HAZARD_T})

$RR_4 \equiv R_{31}$

Tuples for this relationship are shown below. Note that these tuples are the same as the tuples for the DISPOSES relationship from Chapter IV, except that the values for WASTE_NO have been replaced with the equivalent values for BY_PROD_ID.

WDISPOSER_ID	BY_PROD_ID	QTY_DISPOSED	DISPOSAL_HAZARD
WD10	BP002	99	.8
WD30	BP004	35	1
WD30	BP001	25	.5
WD40	BP001	25	.75
WD40	BP002	15	.8

TABLE V-6: DISPOSES_T relationship data

In general, when one-to-one equivalence mappings between entity sets are established, the relationships in which these entities are involved will be transferred to the other entity. All of the relationships between entities in different operational databases that are transferred as a result of one-to-one equivalence mappings are illustrated in Figure V-10 on the next page.

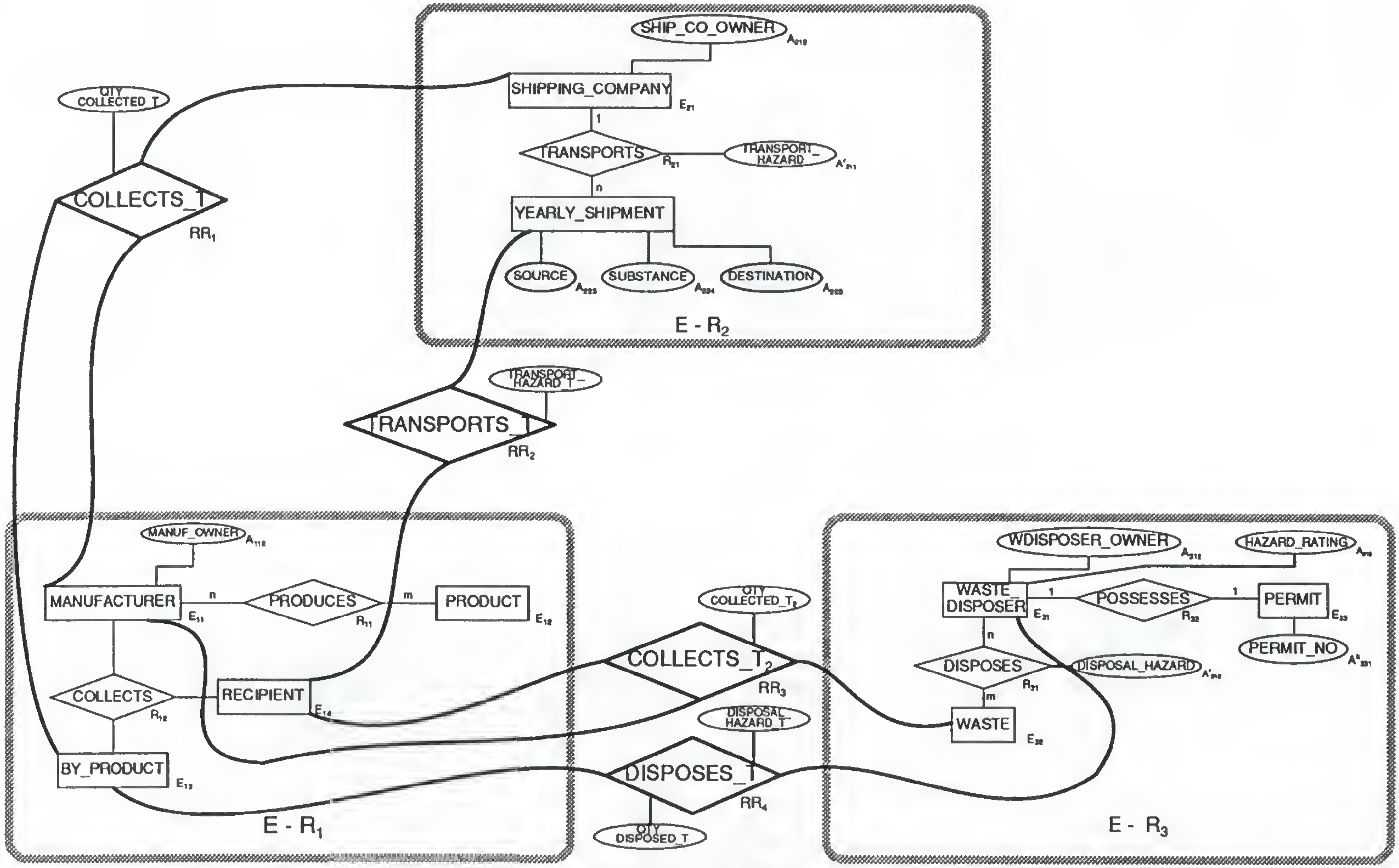


FIG. V-10: RELATIONSHIPS TRANSFERRED AS A RESULT OF ONE-TO-ONE ENTITY EQUIVALENCE MAPPINGS

One-to-one entity projection mappings

Relationships cannot always be transferred if the mapping is a one-to-one projection mapping or a relationship-attribute-to-entity mapping. Relationships apply to whole entities and may not hold if only attributes of entities are being mapped; the DBA must determine whether these relationships hold for the entities involved.

First consider the one-to-one projection mapping

BY_PRODUCT ≡ SUBSTANCE $E_{13} \equiv A_{224}$.

The fact that a BY_PRODUCT corresponds to a SUBSTANCE contained in a YEARLY_SHIPMENT implies a new relationship between the BY_PRODUCT and YEARLY_SHIPMENT entities. This relationship will not be transferred from an existing real relationship because there is no YEARLY_SHIPMENTS counterpart in the MANUFACTURING ORGANIZATION'S database and no BY_PRODUCT counterpart in the SHIPPING ORGANIZATION'S database. Data for this relationship must be derived, and the DBA must specify the procedure for deriving this data. For instance, for every YEARLY_SHIPMENTS entity that contains a certain value for SUBSTANCE, a tuple will be created which contains the SHIPMENT_NO of that shipment and the BY_PROD_ID that corresponds to the SUBSTANCE being shipped.

$RR_5 = (\{A^k_{221}; A^k_{131}\};\{\})$
CONTAINS ({SHIPMENT_NO; BY_PROD_ID};{})
PROCEDURE: for every YEARLY_SHIPMENTS entity that contains a certain value for SUBSTANCE, a tuple will be created which contains the SHIPMENT_NO of that shipment and the BY_PROD_ID that corresponds to the SUBSTANCE being shipped.

Tuples for this new relationship are shown below.

SHIPMENT_NO	BY_PROD_ID
10	BP004
20	BP004
30	BP002
40	BP002
50	BP001
60	BP001

TABLE V-7: CONTAINS relationship data

A similar relationship can be derived from the projection mapping

WASTE ≡ SUBSTANCE $E_{32} \equiv A_{224}$

$RR_6 = (\{A^k_{221}; A^k_{321}\};\{\})$

CONTAINS2 ({SHIPMENT_NO; WASTE_NO};{})

PROCEDURE: for every YEARLY_SHIPMENTS entity that contains a certain value for SUBSTANCE, a tuple will be created which contains the SHIPMENT_NO of that shipment and the WASTE_NO that corresponds to the SUBSTANCE being shipped.

Tuples for this relationship are shown below.

SHIPMENT_NO	WASTE_NO
10	160
20	160
30	190
40	190
50	180
60	180

TABLE V-8: CONTAINS2 relationship data

Consider the third projection mapping.

MANUFACTURER ≡ SOURCE $E_{11} \equiv A_{223}$

The fact that a MANUFACTURER corresponds to a SOURCE contained in a YEARLY_SHIPMENT implies a new relationship between the MANUFACTURER and YEARLY_SHIPMENT entities. This relationship will not be transferred from an existing real relationship because there is no YEARLY_SHIPMENTS counterpart in the MANUFACTURING ORGANIZATION'S database and no SOURCE counterpart in the SHIPPING ORGANIZATION'S database. Data for this relationship must be derived, and the DBA must specify the procedure for deriving this data. For instance, for every YEARLY_SHIPMENTS entity that contains a certain value for SOURCE, a tuple will be created which contains the SHIPMENT_NO of that shipment and the MANUF_ID that corresponds to the SOURCE of the shipment.

$RR_7 = (\{A^k_{111}; A^k_{221}\};\{\})$

SHIPS = ({MANUF_ID; SHIPMENT_NO};{})

PROCEDURE: for every YEARLY_SHIPMENTS entity that contains a certain value for SOURCE, a tuple will be created which contains the SHIPMENT_NO of that shipment and the MANUF_ID that corresponds to the SOURCE of the shipment.

SHIPMENT_NO	MANUF_ID
10	M100
20	M250
30	M050
40	M050
50	M200
60	M200

TABLE V-9: SHIPS relationship data

Similarly, a new relationship is introduced by the mapping
WASTE DISPOSER ≡ DESTINATION $E_{31} \equiv A_{225}$.

$RR_8 = (\{A^k_{311};\ A^k_{221}\};\{\})$
RECEIVES = ({WDISPOSER_ID; SHIPMENT_NO};{})

PROCEDURE: for every YEARLY_SHIPMENTS entity that contains a certain value for DESTINATION, a tuple will be created which contains the SHIPMENT_NO of that shipment and the WDISPOSER_ID that corresponds to the DESTINATION of the shipment.

SHIPMENT_NO	WDISPOSER_ID
10	WD30
20	WD30
30	WD40
40	WD10
50	WD40
60	WD30

TABLE V-10: RECEIVES relationship data

All of these relationships derived as a result of one-to-one entity projection mappings are illustrated in Figure V-11 on the next page.

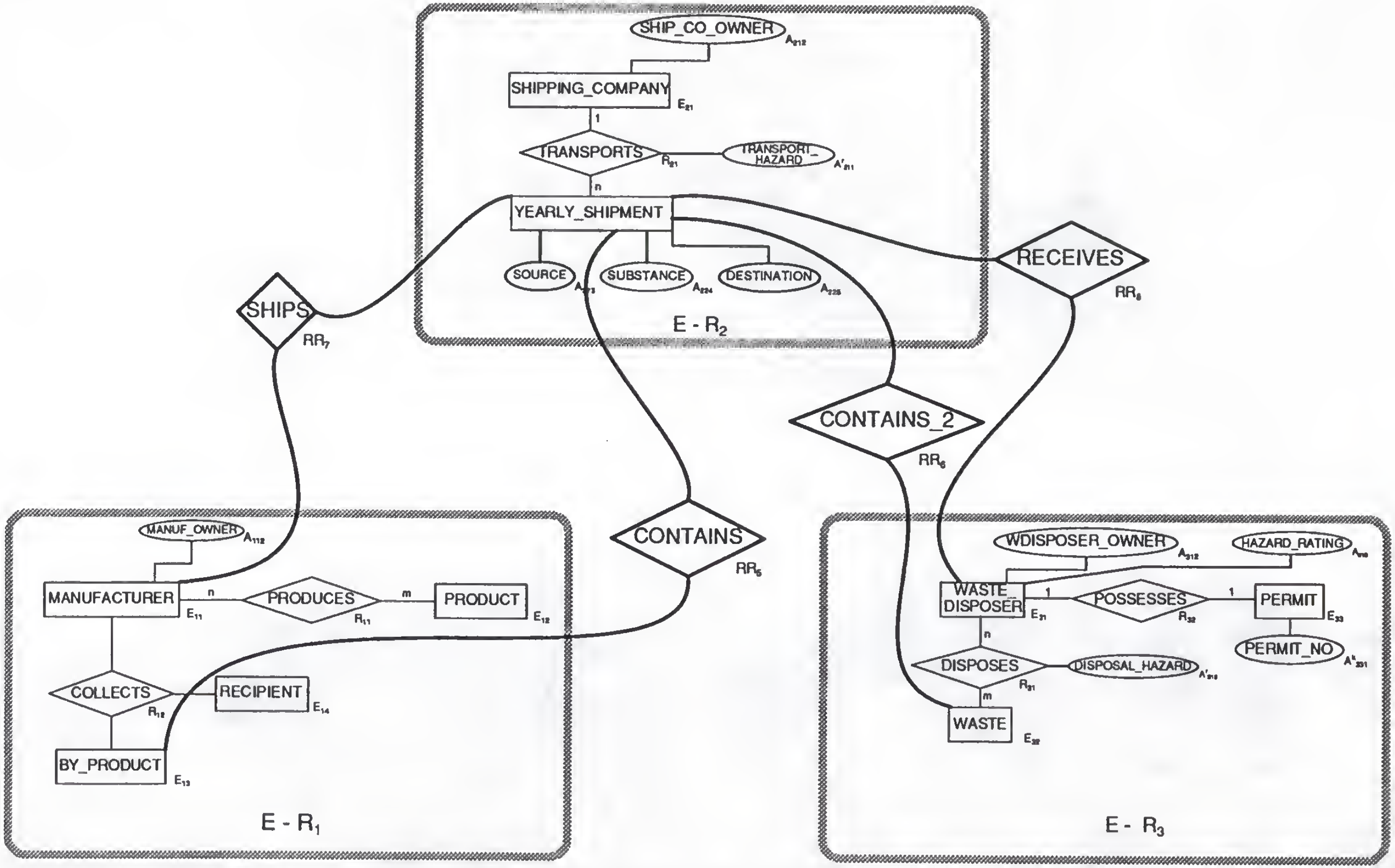


FIG. V-11: RELATIONSHIPS DERIVED AS A RESULT OF ONE-TO-ONE ENTITY PROJECTION MAPPINGS

One-to-many entity mappings

Although there were no one-to-many mappings between real entities in our example, these mappings can also introduce new relationships. For instance, if the entity sets involved in the "many" side of the mapping are not already related, the fact that they are mapped to the same entity set may introduce a new relationship. This new relationship is illustrated in Figure V-12 below.

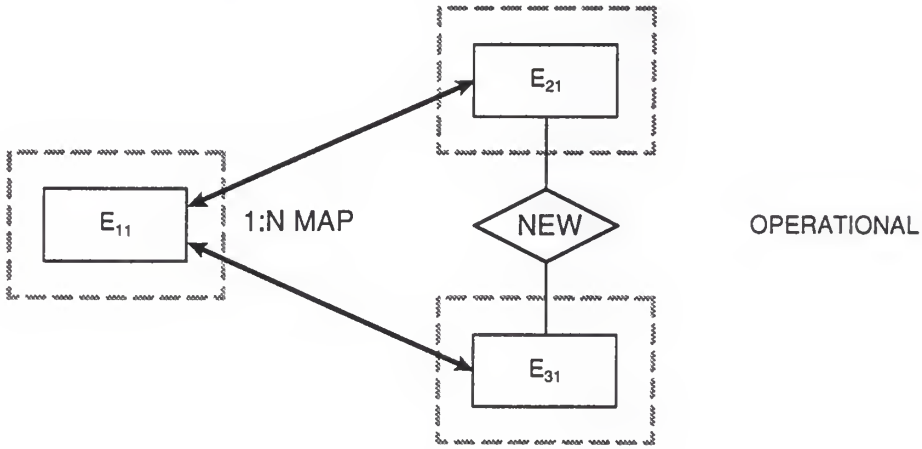


FIG. V-12: POTENTIAL NEW RELATIONSHIP AS A RESULT OF A ONE-TO-MANY ENTITY MAPPING

One-to-one entity attribute mappings

Consider the one-to-one entity attribute mapping

MAN_OWNER ≡ SHIP_CO_OWNER $A_{112} \equiv A_{212}$.

This mapping indicates that the two entities, MANUFACTURER AND SHIPPING_COMPANY, both contain attributes which indicate the owner of the business enterprise. There is no implication that the two enterprises may be held by the same owner, although such a case will be indicated if the two owner values are equivalent. No relationship between the two entities is introduced by this mapping or by the mappings involving the owners of the entities given below.

MAN_OWNER ≡ WDISPOSER_OWNER $A_{112} \equiv A_{312}$
SHIP_CO_OWNER ≡ WDISPOSER_OWNER $A_{212} \equiv A_{312}$

The equivalence between the HAZARD_RATING attribute of a SHIPPING_COMPANY and the HAZARD_RATING attribute of a WASTE_DISPOSER is given below.

HAZARD_RATING ≡ HAZARD_RATING $\qquad A_{213} \equiv A_{313}$

This equivalence is due to the fact that the EPA assigns a rating to all handlers of hazardous waste facilities and both shipping companies and waste disposal facilities will possess such a rating. However, this attribute does not indicate an additional relationship between the involved entities.

Relationship-attribute-to-relationship-attribute mappings

Consider the relationship-attribute-to-relationship-attribute mapping

TRANSPORT_HAZARD ≡ DISPOSAL_HAZARD $\qquad A^r{}_{211} \equiv A^r{}_{311}$.

This equivalence is a result of the fact that the relationships involve facilities which handle hazardous waste and wastes disposed or shipped via a shipment. The interaction between the facility is assigned a hazard rating which depends not only upon the toxicity of the waste but upon the characteristics of the handler as well. Therefore, each of these relationship attributes represents a similar characteristic.

Because a relationship attribute's value is determined by a combination of values of entity key attributes, its equivalence to a component of another database may introduce new relationships between the involved entities. Therefore, this mapping will be respecified in terms of the entity keys.

f(SHIPPING_COMPANY, SHIPMENT) ≡ g(WASTE_DISPOSER, WASTE)

The fact that a hazard rating has been assigned to the interaction between these waste handlers and wastes does not introduce an additional relationship between these entities.

Relationship attribute to entity mappings and
Relationship attribute to entity attribute mappings

Because a relationship attribute's value is determined by a combination of values of entity key attributes, its equivalence to a component of another database may introduce new relationships between the involved entities. Therefore, potential interactions between the entities involved in the relationships and in the mapping should be analyzed. There are no examples of these two mapping types in the example decision problem.

All of the eligible mappings from Step 1 have been examined and all potential relationships have been identified between entities in different databases that resulted from these mappings. However, the DBA may be able to identify additional relationships that are not a direct result of those mappings.

Consider the original purpose for establishing the DSS database in the example decision problem. The objective was to trace hazardous waste from its original generation point to its final destination. While the MANUFACTURER database provides information concerning the transfer of waste from the original manufacturer to a shipment company, there is no established link between the shipping company and the final waste disposal facility. However, we can use information that we have gained from the mappings and from the relationships that cross database boundaries to establish such a link.

Now that the RECEIVES relationship (RR_8) between a WASTE_DISPOSER and YEARLY SHIPMENT has been established, a SHIPMENT can be traced from a SHIPPING_COMPANY via TRANSPORTS to a WASTE DISPOSER via RECEIVES. In addition, because the CONTAINS2 relationship (RR_6) indicates whether a particular WASTE was included in a SHIPMENT, the three relationships can be used to trace the fact that a given WASTE was transferred from a SHIPPING_COMPANY to a WASTE_DISPOSER. Because of the importance of this transfer of waste to the DSS problem domain, this new relationship will be developed.

RR_9 ({A^k_{311}; A^k_{321}; A^k_{211}}; {A^{rr}_{21}})
DELIVERS({WDISPOSER_ID; WASTE_NO; SHIP_CO_ID}; {QTY_DELIVERED})
PROCEDURE: JOIN CONTAINS2 and RECEIVES on SHIPMENT_NO.
JOIN THE RESULT WITH TRANSPORTS on SHIPMENT_NO.
The result of this part of the procedure is given below.

WASTE_NO	SHIP_CO_ID	WDISPOSER_ID	SHIPMENT_NO
160	SC20	WD30	10
160	SC20	WD30	20
190	SC05	WD40	30
190	SC05	WD10	40
180	SC10	WD40	50
180	SC10	WD30	60

TABLE V-11(a): INTERMEDIATE STEP IN DEVELOPMENT OF DELIVERS RELATIONSHIP

Because the SHIPMENT_NO key attribute functionally determines the non-key attribute QUANTITY of the YEARLY_SHIPMENTS entity, and the QUANTITY attribute is semantically equivalent to the QTY_DELIVERED attribute of this relationship, add the QUANTITY attribute to the tuples, calling it QTY_DELIVERED.

WASTE_NO	SHIP_CO_ID	WDISPOSER_ID	SHIPMENT_NO	QTY_DELIVERED
160	SC20	WD30	10	10
160	SC20	WD30	20	25
190	SC05	WD40	30	15
190	SC05	WD10	40	99
180	SC10	WD40	50	25
180	SC10	WD30	60	25

TABLE V-11(b): INTERMEDIATE STEP IN DEVELOPMENT OF DELIVERS RELATIONSHIP

Finally, because the SHIPMENT_NO is not of interest in this relationship, we wish to project out the SHIPMENT_NO but still retain the quantity delivered information. A simple projection would result in multiple tuples with the same values for WASTE_NO, SHIP_CO_ID, and WDISPOSER_ID; for these multiple tuples, the QTY_DELIVERED attribute will be summed and duplicates eliminated. This part of the procedure produces the final tuples for the relationship DELIVERS, as shown below.

WASTE_NO	SHIP_CO_ID	WDISPOSER_ID	QTY_DELIVERED
160	SC20	WD30	35
190	SC05	WD40	15
190	SC05	WD10	99
180	SC10	WD40	25
180	SC10	WD30	25

TABLE V-11(c): FINAL STEP IN DEVELOPMENT OF DELIVERS RELATIONSHIP

TABLE V-11: DEVELOPMENT OF TUPLES FOR THE DELIVERS RELATIONSHIP

This notation and discussion above indicate that real entities E_{31}, WASTE DISPOSER, E_{32}, WASTE, and E_{21}, SHIPPING COMPANY, are related. The relationship between these three entities also generates its own attribute, $A^{rr}{}_{21}$, which we will call QTY_DELIVERED.

This relationship, which is not a direct result of mappings at the operational level, is illustrated in Figure V-13 on the next page.

All of the relationships between entities in different operational databases are shown in Figure V-14 on the following page.

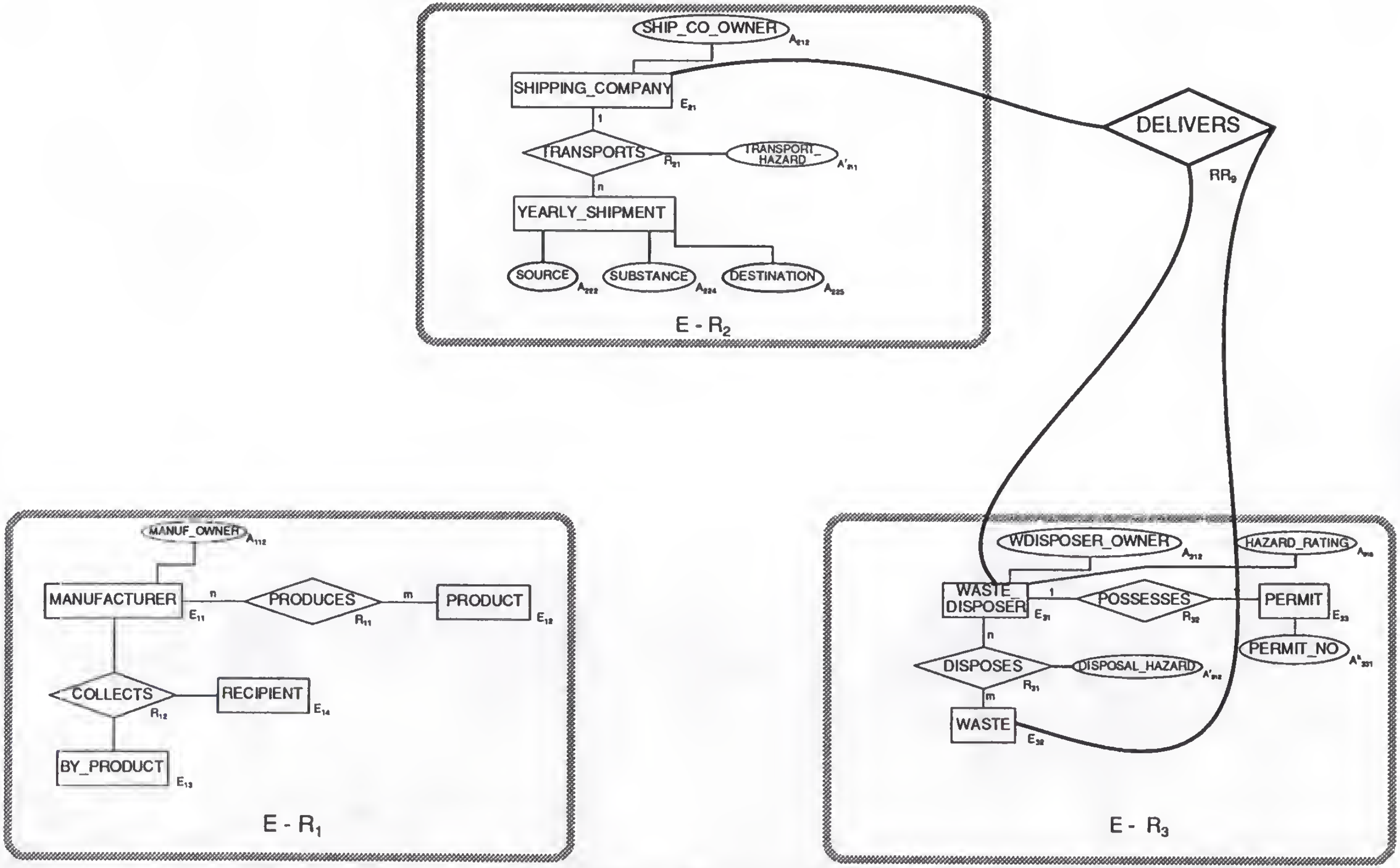


FIG. V-13: RELATIONSHIP WHICH IS NOT A DIRECT RESULT OF MAPPINGS AT THG OPERATIONAL LEVEL

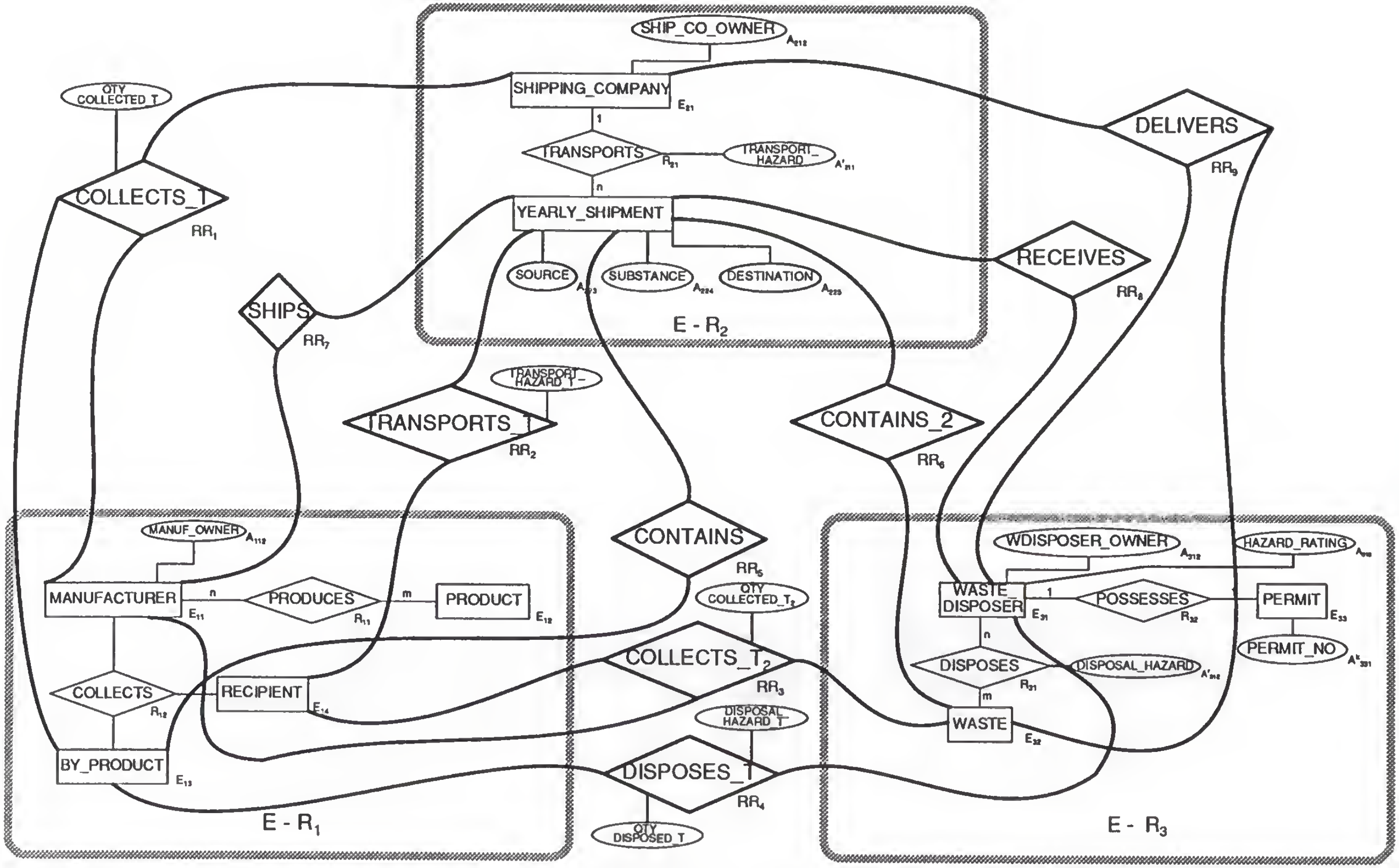


FIG. V-14: ALL RELATIONSHIPS BETWEEN ENTITIES IN DIFFERENT OPERATIONAL DATABASES

New relationships should be carefully analyzed by the DBA as potential virtual relationships. With all of the semantic equivalences between the entities specified and relationships between them identified, the next step is to form mappings between the operational and virtual levels.

B.3.3 (STEP 3) IDENTIFY MAPPINGS FROM REAL TO VIRTUAL ENTITY SETS

In the previous steps, the semantic equivalences between components of the different operational databases were specified through mappings, and relationships which cross database boundaries were identified. The next step is to identify mappings to the virtual level, which are shown using dotted lines in Figure V-15 below.

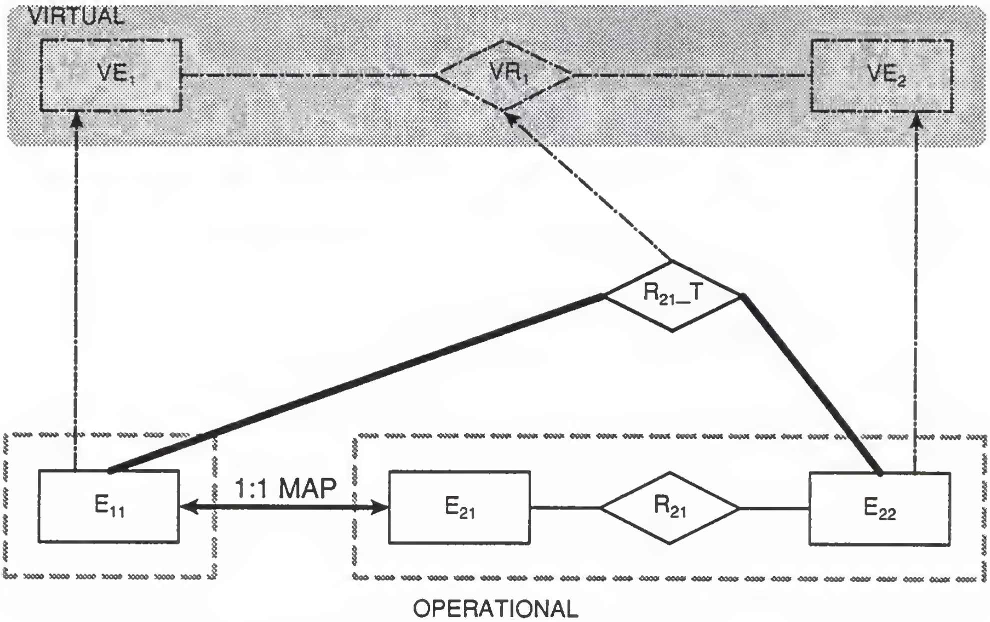


FIG. V-15: OPERATIONAL TO VIRTUAL LEVEL MAPPINGS

The subjects of this section are mappings between the operational and virtual levels, and we begin with mappings from real to virtual entity sets. The objective at this stage is to select virtual entities that will provide the user with a natural representation of the information in the operational databases, given the problem domain. We have specified equivalences between different components of different databases so that information describing the same physical object can be identified easily.

Real entity sets and relationships that are not included as part of the virtual E-R will not be a part of the user's view of the data available from the DSS database. However, these excluded real entities and relationships may be used to derive information that becomes a part of the virtual E-R. Mappings from the operational to the virtual level should include all equivalences, since this information will be used to retrieve information from the operational databases.

Which entities are included at the virtual level and how they are formed will be determined by the DBA using the information gathered in Steps 1 and 2. Points to be considered in making these decisions are outlined in the next section.

B.3.3.1 USING INFORMATION FROM THE PREVIOUS STEPS TO EVALUATE POTENTIAL VIRTUAL ENTITIES

This stage involves the evaluation of the information available from the mappings at the operational level and the relationships between real entities in different operational databases. The objective of this analysis is to determine the virtual entities that will represent the user's view of information in the DSS. We will specify how the virtual entities are formed through a series of mappings.

The different types of possible mappings are illustrated in Figure V-16 below. As with real entity mappings, we will not consider many-to-many entity set mappings to the virtual level.

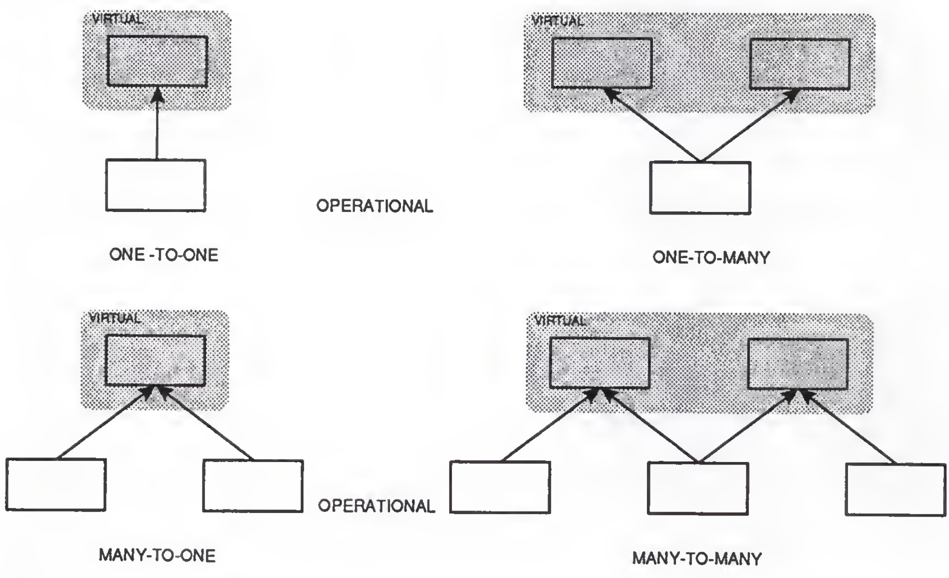


FIG. V-16: TYPES OF MAPPINGS BETWEEN OPERATIONAL AND VIRTUAL ENTITY SETS

B.3.3.1.1 FORMING VIRTUAL ENTITY SETS WHICH HAVE REAL ENTITY SET COUNTERPARTS

In considering whether to form a single virtual entity set from a real entity set, consider the following procedure:

(1) Is the real entity set of critical importance to the problem situation?

If not, skip to question 3.
Otherwise continue.

(2) Is the real entity set involved in any mappings with entity sets from different operational databases?

If not, it can be mapped directly to the virtual level. Skip to question 3.

If yes, continue.

(2a) Is the entity involved in a one-to-one equivalence mapping?

If yes, one virtual entity set can represent all of the real entity sets from the mapping. Attribute mappings will be necessary to determine how the virtual entity attributes will be established. Three alternatives are possible:

(i) Take the union of all attributes in the two real entity sets.

(ii) Choose the attributes of one of the entity sets to be sufficient coverage for all attributes of all entity sets. This choice will be made based upon the DBA's knowledge of the problem domain.

(iii) Combine sets of attributes so that if more than one entity attribute describes the same physical characteristic of the virtual entity, only one of these attributes is mapped to the virtual level. If only one attribute describes a physical characteristic, map it to the virtual level.

If not, continue.

(2b) Is the real entity set involved in a one-to-one projection mapping with a second entity set?

If yes, is the second entity set of interest to the problem domain?

If yes, both of the entity sets should become entities at the virtual level. However, there are two alternatives for mapping the second entity set:

(i) Map the complete entity set to the virtual level.
(ii) Map only the attributes of the second entity set which are not involved in the projection.

The DBA must determine which mapping is most suitable for the problem domain.

If not, only the first entity set will become an entity at the virtual level.

If not, continue.

(2c) Is the real entity involved in a one-to-many mapping with other entities from operational databases?

If yes, four alternatives exist:

(i) Choose the "1" entity to represent all N+1 entities.
(ii) Choose m <= N+1 entities to represent all N+1 entities
(iii) Map N+1 entities as separate entities
(iv) Create one concatenated entity that will represent all N+1 entities. If this choice is selected, there are two alternatives:
  (a) Take the union of all attributes in the N+1 entities.
  (b) Some attributes may overlap, so that each attribute must be mapped individually from each of the N+1 entities.

The DBA must determine which mapping is most suitable for the problem domain.

If not, continue.

In making the decision whether to create a concatenated entity set from two or more real entity sets, the relationships involving the real entities should be considered. Consider a case in which we wish to combine two entity sets into a single entity set at the virtual level, and entities from each of these entity sets is involved in relationships with entities from other entity sets. Relationships between the new "combined" entity and the other entities may be very complex. This information should be factored into the decision whether to form the combined entity set and which of the relationship to maintain at the virtual level. This point is illustrated in Figure V-17 below.

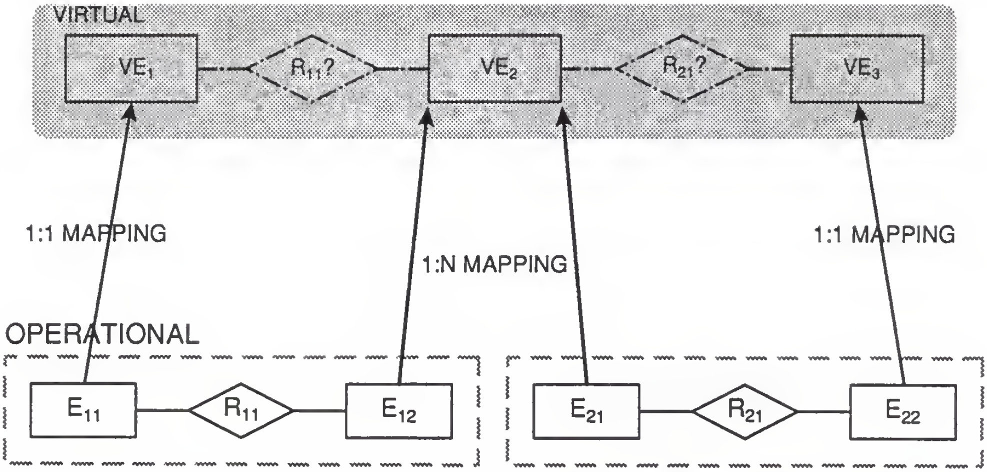


FIG. V-17: POTENTIAL VIRTUAL RELATIONSHIPS AS A RESULT OF A MANY-TO-ONE ENTITY MAPPING TO THE VIRTUAL LEVEL.

(3) Is the entity set involved in any real relationships or relationships with entity sets from different operational databases? If so, are these entity sets being mapped to the virtual level? Do the relationships provide information that will be useful in the problem domain?

The procedure specified above will enable the DBA to identify real entities which are of interest at the virtual level.

B.3.3.1.2 FORMING VIRTUAL ENTITY SETS FROM ENTITY OR RELATIONSHIP ATTRIBUTES

In considering whether to form a virtual entity set which does not have an exact real entity set counterpart (but which would be derived from either an entity attribute or relationship attribute), consider:

(1) Would the information gained be critical to the problem situation?
(2) Would the new virtual entity be related to other virtual entities being created?
(3) Is the DSS to provide information about the virtual entity other than that contained in the entity or relationship attribute? If so, where will the additional information about the virtual entity be obtained and how will it be combined with the existing operational database information?

B.3.3.1.3 An Example of the Analysis for Determining Virtual Entities

This section outlines the above analysis for the example decision problem. In the analysis for the example databases, we are interested in tracing the transfers of a particular WASTE from its original GENERATOR, to any TRANSPORTER, to the final WASTE DISPOSER. We are also interested in identifying business OWNERS which are involved in more than one phase of the waste-handling chain.

We must consider each of the real entities in the operational databases for mapping to the virtual level:

Operational database	Entity		Primary key	
NATIONAL MANUFACTURING ORGANIZATION E-R_1	MANUFACTURER PRODUCT BY_PRODUCT RECIPIENT	E_{11} E_{12} E_{13} E_{14}	MANUF_ID PROD_ID BY_PROD_ID RECIP_ID	A^k_{111} A^k_{121} A^k_{131} A^k_{141}
NATIONAL SHIPPING ORGANIZATION E-R_2	SHIPPING_COMPANY YEARLY_SHIPMENTS	E_{21} E_{22}	SHIP_CO_ID SHIPMENT_NO	A^k_{211} A^k_{221}
EPA E-R_3	WASTE DISPOSER WASTE PERMIT	E_{31} E_{32} E_{33}	WDISPOSER_ID WASTE_NO PERMIT_NO	A^k_{311} A^k_{321} A^k_{331}

TABLE V-1: REAL ENTITIES IN EXAMPLE DATABASES (REPEATED)

Since this first step determines the virtual entities which have real entity set counterparts, we must also consider the following mappings, since they involve whole entity sets.

One-to-one entity equivalence mappings:	
BY_PRODUCT ≡ WASTE RECIPIENT ≡ SHIPPING COMPANY	$E_{13} \equiv E_{32}$ $E_{14} \equiv E_{21}$
One-to-one projection mappings:	
BY_PRODUCT ≡ SUBSTANCE WASTE ≡ SUBSTANCE MANUFACTURER ≡ SOURCE WASTE_DISPOSER ≡ DESTINATION	$E_{13} \equiv A_{224}$ $E_{32} \equiv A_{224}$ $E_{11} \equiv A_{223}$ $E_{31} \equiv A_{225}$.
One-to-many entity mappings: none	
Relationship-attribute-to-entity mappings: none	

TABLE V-12: EXAMPLE MAPPINGS INVOLVING WHOLE ENTITIES

Each real entity set will be evaluated in turn.

MANUFACTURER:
(1) is of importance to the problem situation
(2) is involved in mappings with entity sets from different operational databases:
   (2a) is not involved in one-to-one equivalence mappings.
   (2b) is involved in one-to-one projection mapping with SOURCE, which is an attribute of a YEARLY_SHIPMENT. The relationship between MANUFACTURER and YEARLY_SHIPMENT is of interest only because it involves the transfer of waste from the MANUFACTURER to another waste handler.
   (2c) is not involved in one-to-many entity mappings.
(3) is involved in relationships with other entities: PRODUCT, RECIPIENT, BY_PRODUCT, SHIPPING_COMPANY, and YEARLY_SHIPMENT.

PRODUCT:
(1) is not of importance to the problem situation

BY-PRODUCT:
(1) is of importance to the problem situation
(2) is involved in mappings with entity sets from different operational databases:
   (2a) is involved in one-to-one equivalence mapping with WASTE. Therefore, one virtual entity can represent both BY_PRODUCT and WASTE. If this option is chosen, the DBA must determine attribute mappings.
   (2b) is involved in one-to-one projection mapping with SUBSTANCE, which is an attribute of YEARLY_SHIPMENT. The relationship between BY_PRODUCT and YEARLY_SHIPMENT is of interest only because it involves the transfer of the BY_PRODUCT from the MANUFACTURER to another waste handler.
   (2c) is not involved in one-to-many entity mappings.
(3) is involved in relationships with other entities: MANUFACTURER, RECIPIENT, YEARLY_SHIPMENT, SHIPPING_COMPANY, and WASTE_DISPOSER.

RECIPIENT:
(1) is of importance to the problem situation
(2) is involved in mappings with entity sets from different operational databases:
   (2a) is involved in one-to-one equivalence mapping with SHIPPING_COMPANY.
   Therefore, one virtual entity can represent both RECIPIENT and SHIPPING_COMPANY. If this option is chosen, the DBA must determine attribute mappings.
   (2b) is not involved in one-to-one projection mappings.
   (2c) is not involved in one-to-many entity mappings.
(3) is involved in relationships with other entities: MANUFACTURER, BY_PRODUCT, WASTE, and YEARLY_SHIPMENT.

SHIPPING COMPANY:

(1) is of importance to the problem situation

(2) is involved in mappings with entity sets from different operational databases:

   (2a) is involved in one-to-one equivalence mapping with RECIPIENT. Therefore, one virtual entity can represent both RECIPIENT and SHIPPING_COMPANY. If this option is chosen, the DBA must determine attribute mappings.

   (2b) is not involved in one-to-one projection mappings.

   (2c) is not involved in one-to-many entity mappings.

(3) is involved in relationships with other entities: MANUFACTURER, BY_PRODUCT, WASTE, YEARLY_SHIPMENT, and WASTE_DISPOSER.

YEARLY SHIPMENTS:

(1) is of interest to the problem domain only because it involves the transport of WASTE between sites; because this information can be represented using other entities and relationships, this entity will not be mapped to the virtual level.

WASTE DISPOSER:

(1) is of importance to the problem situation

(2) is involved in mappings with entity sets from different operational databases:

   (2a) is not involved in one-to-one equivalence mappings.

   (2b) is involved in one-to-one projection mapping with DESTINATION, which is an attribute of YEARLY_SHIPMENT. The relationship between WASTE_DISPOSER and YEARLY_SHIPMENT is of interest only because it involves the transfer of the waste from the SHIPPING_COMPANY to a WASTE_DISPOSER.

   (2c) is not involved in one-to-many entity mappings.

(3) is involved in relationships with other entities: WASTE, PERMIT, SHIPPING_COMPANY, YEARLY_SHIPMENT, MANUFACTURER, and RECIPIENT.

It will be of interest, for the example problem situation, to identify whether a particular WASTE_DISPOSER facility is permitted. Therefore, at the virtual level, this entity will be augmented by adding permit information.

WASTE:

(1) is of importance to the problem situation

(2) is involved in mappings with entity sets from different operational databases:

   (2a) is involved in one-to-one equivalence mapping with BY_PRODUCT. Therefore, one virtual entity can represent both BY_PRODUCT and WASTE. If this option is chosen, the DBA must determine attribute mappings.

   (2b) is involved in one-to-one projection mapping with SUBSTANCE, which is an attribute of YEARLY_SHIPMENT. The fact that a WASTE is included in a shipment from one waste handler to another is the only information of interest concerning a YEARLY_SHIPMENT.

(2c) is not involved in one-to-many entity mappings.

(3) is involved in relationships with other entities: WASTE_DISPOSER, MANUFACTURER, RECIPIENT, YEARLY_SHIPMENT, and SHIPPING_COMPANY.

PERMIT:

(1) by itself, is not of importance to the problem situation. However, whether a waste disposal facility is permitted is of interest to the problem situation.

Therefore, we have identified four virtual entities which have real entity counterparts: WASTE, which represents BY_PRODUCT and WASTE; GENERATOR, which represents MANUFACTURER; TRANSPORTER, which represents SHIPPING_COMPANY and RECIPIENT; and WASTE DISPOSER, which represents WASTE DISPOSER. Real entities which will not be represented at the virtual level as entities include PRODUCT, which is not of interest to the problem situation; PERMIT, which is of interest only as an attribute of a WASTE_DISPOSER; and YEARLY_SHIPMENT, which will be represented indirectly by relationships that associate entities that transfer waste.

We must now examine the mappings which involve entity and relationship attributes for other potential virtual entities which do not have real entity counterparts. These mappings will not involve whole entities, but only attributes of relationships or entities.

One-to-one entity attribute mappings:

MAN_OWNER ≡ SHIP_CO_OWNER	$A_{112} \equiv A_{212}$
MAN_OWNER ≡ WDISPOSER_OWNER	$A_{112} \equiv A_{312}$
SHIP_CO_OWNER ≡ WDISPOSER_OWNER	$A_{212} \equiv A_{312}$
HAZARD_RATING ≡ HAZARD_RATING	$A_{213} \equiv A_{313}$

Relationship attribute to relationship attribute mappings:

TRANSPORT_HAZARD ≡ DISPOSAL_HAZARD	$A^r{}_{211} \equiv A^r{}_{312}$

The first three mappings would contribute information about the owners of business units that would be important to the problem situation. A new OWNER entity would be related to the GENERATOR, TRANSPORTER, and WASTE_DISPOSER entities.

The HAZARD RATING attribute serves well as an attribute of another entity. Generally, we are not interested in a HAZARD RATING without knowing the waste handler that it describes. Similarly, information concerning the hazard index for a particular interaction between waste handlers and wastes is not of interest outside that relationship. Therefore, this analysis has resulted in only one virtual entity does not have a real entity counterpart: OWNER.

The result of this analysis is the specification of which virtual entities are to be created and how they will be formed. We have specified the five virtual entities that will be created for the example problem situation; how they will be formed will be specified in the format outlined in the next section.

B.3.3.2 EXAMPLES AND NOTATION FOR MAPPINGS TO VIRTUAL ENTITIES

Virtual entity sets may have a real entity set counterpart, or they may be created as a result of combining information about existing operational database entity sets. Mappings from real entity sets to virtual entity sets will be specified in a manner similar to that used for mappings between real entity sets in different operational databases.

As with mappings at the operational level, a many-to-many mapping would occur whenever information concerning n real entity sets is arranged in the virtual E-R so that it was used to describe m different entity sets. We will assume that many-to-many mappings are rare, and that they may be specified using a collection of one-to-many mappings rather than developing a separate notation for them.

B.3.3.2.1 SPECIFYING VIRTUAL ENTITY SETS

Assume we are interested in specifying the mappings from a collection of real entities in the operational databases to the virtual entities formed from them. This section will explain the mappings that will identify virtual entities and show, when necessary, how attribute sets for the virtual entities are formed.

<u>One-to-one Entity Equivalence Mappings</u>

This section presents the notation for a one-to-one entity equivalence mapping.

(1:1 MAPPINGS) (a) The virtual entity of interest is equivalent to a real entity which exists in one of the $E\text{-}R_p$'s:

$$\mathbf{VE_i \equiv E_{pq}}.$$

$$\therefore \quad VA^k{}_{ij} \equiv A^k{}_{pqr}.$$

$$\text{TRANSLATION FUNCTION:} \quad VA^k{}_{ij} = t(A^k{}_{pqr}).$$

The key value for the virtual entity will be equivalent to the key value(s) for the real entity.

A virtual entity set may have one-to-one equivalence mappings with several real entity sets. Recall that a mapping indicates that the two entity sets represent the same physical object, but that their attributes may describe different characteristics of the entity. All of the real entity sets which are involved in one-to-one equivalence mappings with a virtual entity set must be identified. If only one real entity set is identified, then the virtual entity set can be an exact copy of the real entity set. In our example databases, there is one virtual entity set which illustrates this point, as shown below.

GENERATOR (VIRTUAL) ≡ MANUFACTURER $\qquad VE_1 \equiv E_{11}$
GENERATOR_ID ≡ MANUF_ID $\qquad VA^k{}_{11} = A^k{}_{111}$

All attributes of the virtual GENERATOR entity will be taken directly from the real MANUFACTURER entity; the key value, GENERATOR_ID, will be an exact copy of MANUF_ID.

In the example databases, there are also chosen two virtual entity sets which have one-to-one equivalence mappings with more than one real entity set. In this case, each of the attributes of the virtual entity set must be mapped from the attributes of the equivalent real entity sets. The first of these two virtual entities is WASTE.

WASTE (VIRTUAL) ≡ WASTE $\qquad VE_2 \equiv E_{32}$
WASTE (VIRTUAL) ≡ BY_PRODUCT $\qquad VE_2 \equiv E_{13}$

The key attribute for the WASTE virtual entity must be specified by the DBA; in this case, it will be identical to the key attribute for the WASTE real entity. Recall that for any equivalence mapping, we assume that the key values of each entity being mapped can be translated into key values of the all other entities involved in the mapping. In this case, the look-up table shown in Table V-3 is used to translate BY_PROD_ID values into values for WASTE_NO.

$VA^k{}_{21} = A^k{}_{321} \equiv A^k{}_{131}$
WASTE_NO = WASTE_NO ≡ BY_PROD_ID

The DBA must also determine which attributes the virtual WASTE entity will possess and how they will be initialized. These non-key attributes are taken from the real entities, including the RANKING attributes discussed earlier; note that one virtual entity attribute is mapped for each of the real RANKING attributes.

$VA_{22} = A_{322}$
WASTE_NAME = WASTE_NAME

$VA_{23} = A_{132}$
INFLAMMABILITY_INDEX = RANKING

$VA_{24} = A_{323}$
TOXICITY_SCORE = RANKING

The second example of a virtual entity with more than one equivalent real entity is given below.

TRANSPORTER (VIRTUAL) ≡ SHIPPING COMPANY $VE_3 \equiv E_{21}$
TRANSPORTER (VIRTUAL) ≡ RECIPIENT $VE_3 \equiv E_{14}$

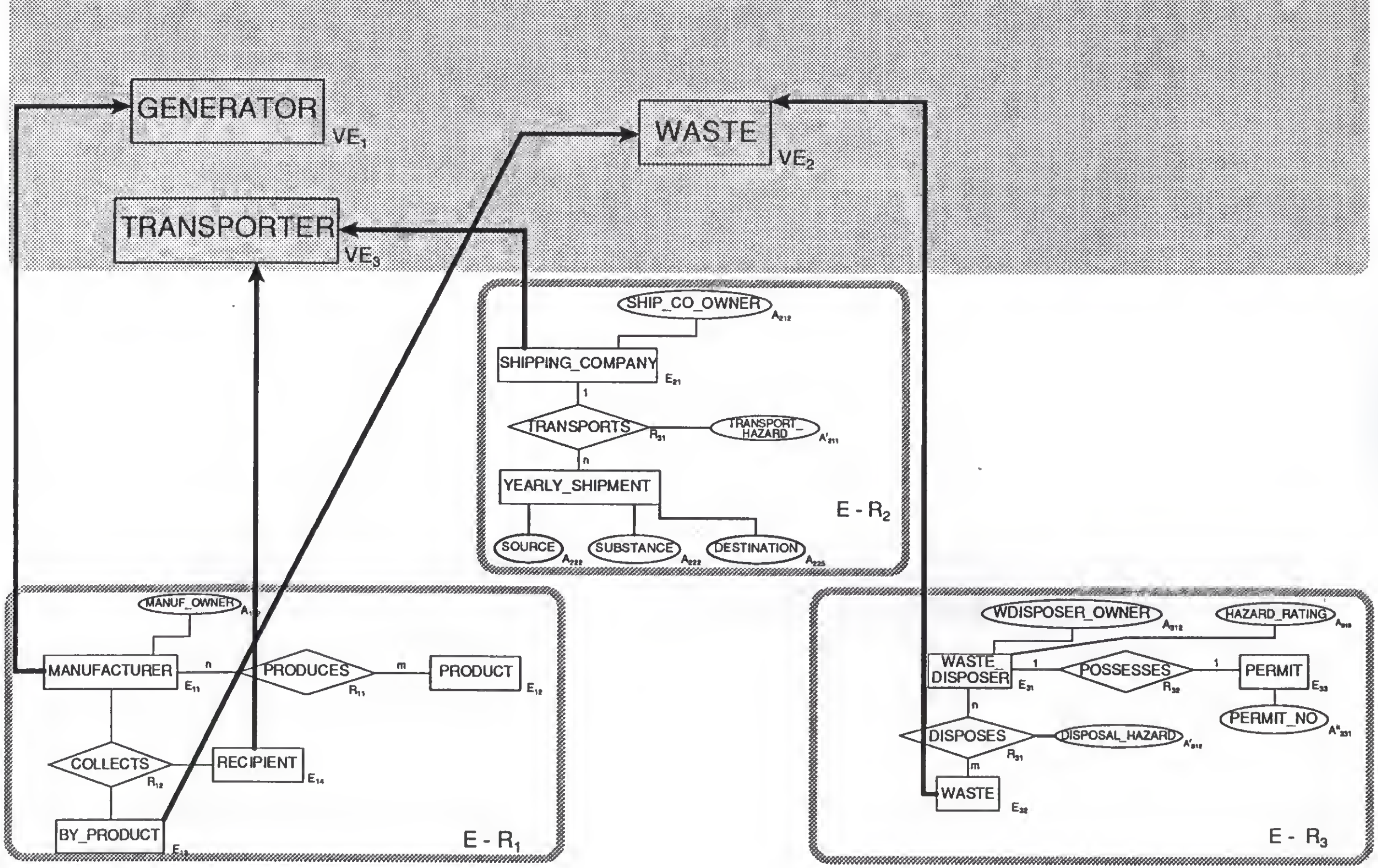
$VA^k{}_{31} = A^k{}_{211} = f(A^k{}_{141})$
TRANSPORTER_ID = SHIP_CO_ID = RECIP_ID

The DBA specifies that the key attribute of the TRANSPORTER entity set will be the SHIP_CO_ID, which has values from the same domain as RECIP_ID. The only other attribute illustrated for SHIPPING_COMPANY and RECIPIENT is the SHIP_CO_OWNER attribute, which will also become an attribute of the TRANSPORTER entity; it will be renamed as TRANSPORTER_OWNER.

$VA_{32} = A_{212}$
TRANSPORTER_OWNER = SHIP_CO_OWNER

Virtual entities formed as a result of one-to-one equivalence mappings are shown in Figure V-18 on the next page.

FIG. V-18: FORMATION OF VIRTUAL ENTITIES: ONE-TO-ONE ENTITY EQUIVALENCE MAPPINGS



One-to-one Entity Projection Mappings

This section presents the notation for a one-to-one entity projection mapping.

(1:1 MAPPINGS) (b) The virtual entity of interest is equivalent to a subset of attributes of a real entity which exists in an $E\text{-}R_p$. In essence, the entity is formed as a projection of the entity E_{pq}:

$$\mathbf{VE_i \equiv \{A_{pqr},\ r=1,n\}}.$$

The key value(s) for the virtual entity will therefore be equivalent to the value(s) for the real attribute(s). Assume that only one attribute is involved in the projection.

$$\therefore \quad VA^k{}_{ij} \equiv A_{pqr}.$$

$$\text{TRANSLATION FUNCTION:} \quad VA^k{}_{ij} = t(A_{pqr}).$$

Any non-key attribute is a function of the value of the entity's key attribute(s).

$$A_{pqr} = f(A^k{}_{pqs}).$$

$$\therefore \quad VA^k{}_{ij} \equiv f(A^k{}_{pqs}).$$

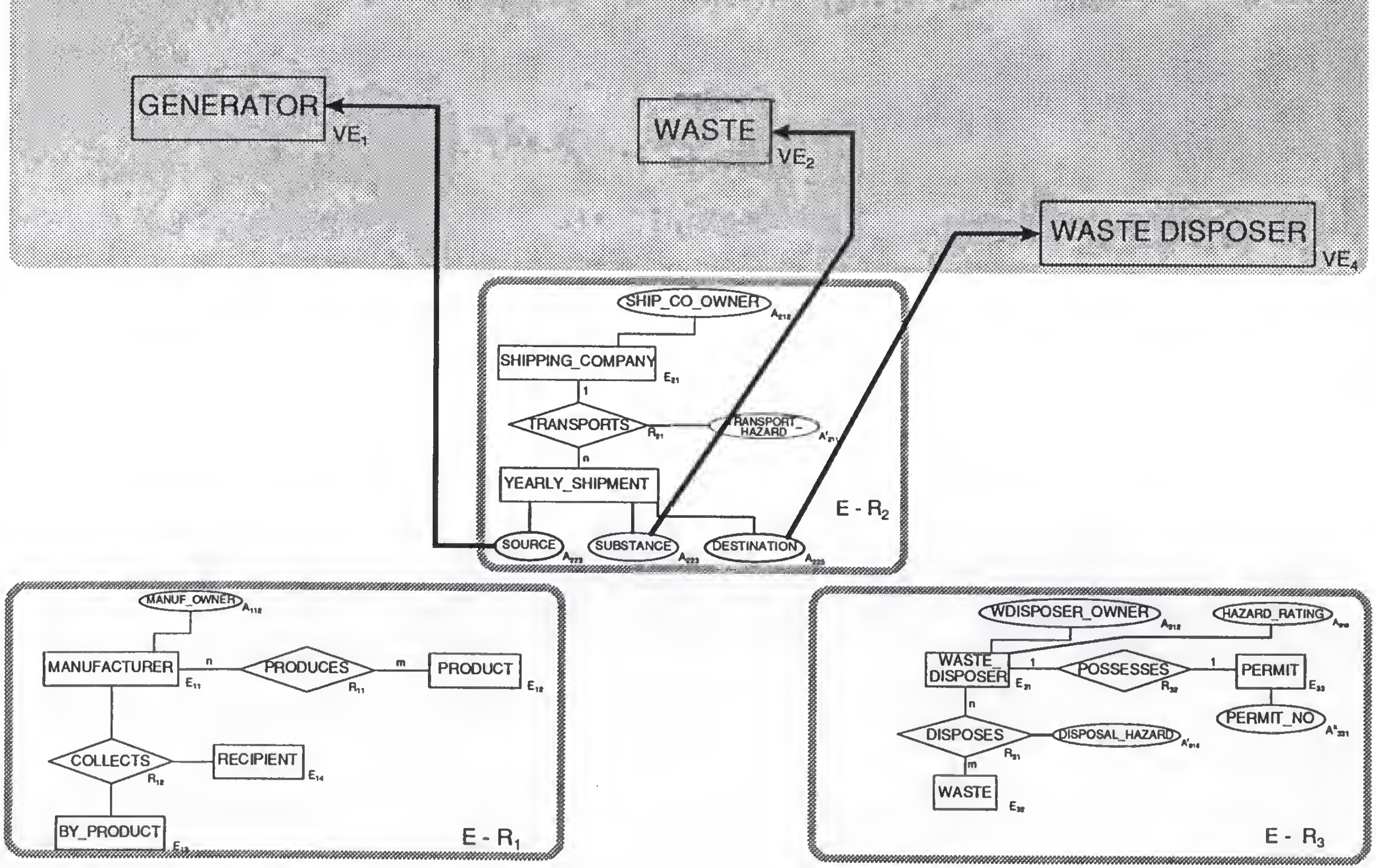
Examples from the problem databases are given below.

WASTE (VIRTUAL) ≡ SUBSTANCE	$VE_2 \equiv A_{224}$
GENERATOR (VIRTUAL) ≡ SOURCE	$VE_1 \equiv A_{223}$
WASTE_DISPOSER (VIRTUAL) ≡ DESTINATION	$VE_4 \equiv A_{225}$
OWNER (VIRTUAL) ≡ MAN_OWNER	$VE_5 \equiv A_{112}$
OWNER (VIRTUAL) ≡ SHIP_CO_OWNER	$VE_5 \equiv A_{212}$
OWNER (VIRTUAL) ≡ WDISPOSER_OWNER	$VE_5 \equiv A_{312}$

WASTE_DISPOSER and OWNER are the only new virtual entity sets, since WASTE and GENERATOR have already been identified as a result of one-to-one equivalence mappings. WASTE_DISPOSER will be discussed further in the next section because it is also the result of a many-to-one entity set mapping. OWNER will be discussed in the section concerning virtual entity sets that do not have a real entity counterpart.

Even though these mappings may not introduce new virtual entities, specification of the mappings is important because they identify additional information about the virtual entities and their relationships to the operational data. The virtual entities involved in one-to-one projection mappings are shown in Figure V-19 on the next page.

FIG. V-19: FORMATION OF VIRTUAL ENTITIES: ONE-TO-ONE ENTITY PROJECTION MAPPINGS



One-to-many Entity Mappings

(1:n MAPPINGS)

(c) The virtual entity of interest consists of a concatenation of real entities or attributes of entities from the different E-R's:

$$VE_i \equiv \prod_{y=1}^{n} X_y \quad \text{where } X_y = E_{ab} \mid A_{pqr}. \text{ and } n \geq 2.$$

$VA^k{}_{ij} \equiv f(\text{keys of } X_{y,\ y=1,n}).$

TRANSLATION FUNCTION: $VA^k{}_{ij} = t(\text{keys of } X_{y,y=1,n}).$

Because of the nature of the problem situation, it would be useful to know whether a particular WASTE_DISPOSER has an EPA permit for handling hazardous waste. However, it is not necessary to have all of the information concerning each permit available at the virtual level. Therefore, we have chosen to modify the WASTE_DISPOSER entity set by adding the PERMIT_NO of the EPA permit possessed by the facility.

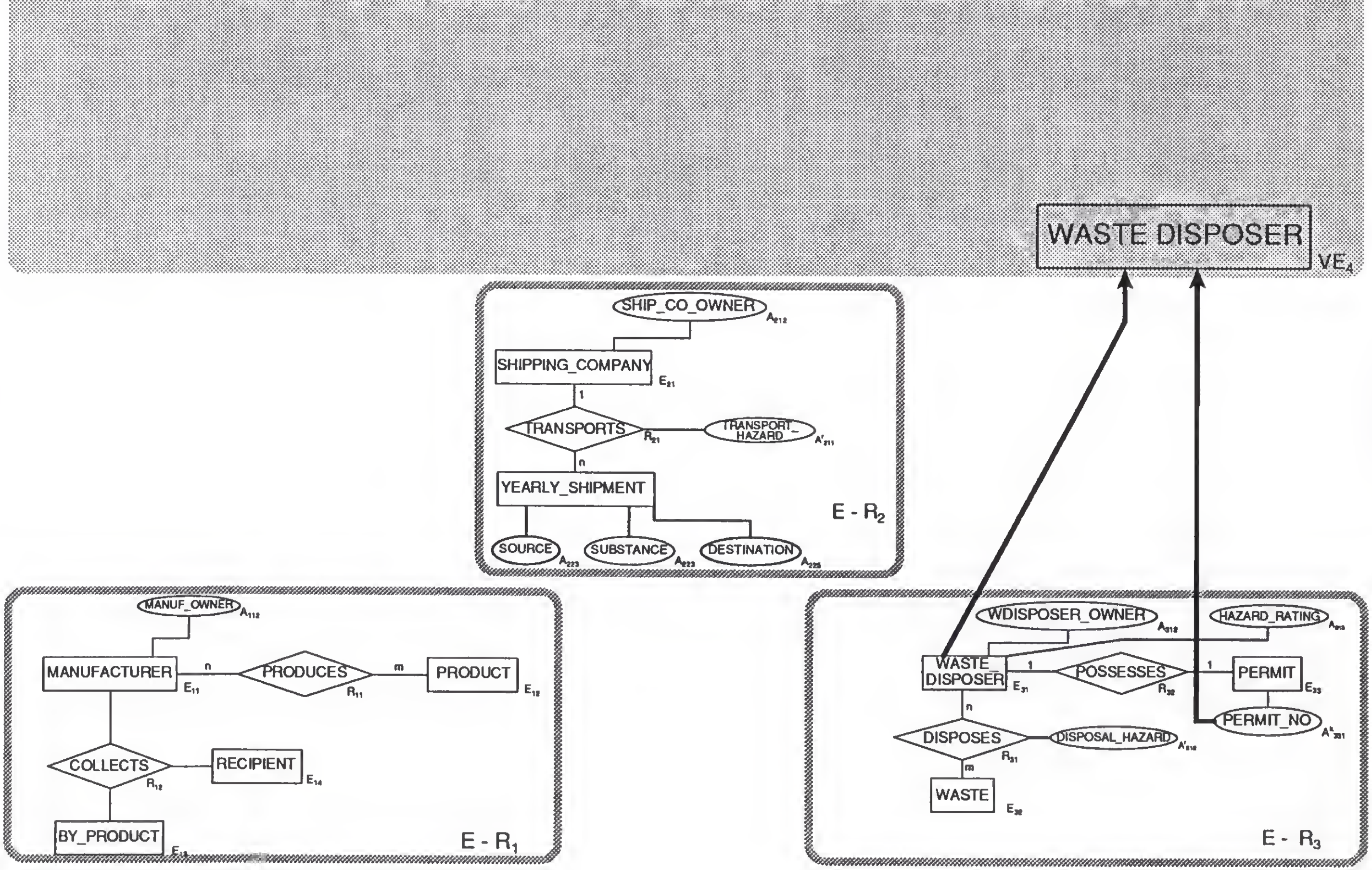
$VE_4 = (E_{31} \mid\mid A^k{}_{331})$
WASTE_DISPOSER (VIRTUAL) = WASTE_DISPOSER || PERMIT_NO

The key of this new virtual entity will be the key of the WASTE entity from which it was created; although PERMIT_NO is the key of the PERMIT real entity, the key of the resulting WASTE_DISPOSER virtual entity will not contain PERMIT_NO.

$VA^k{}_{41} = A^k{}_{311}.$

This one-to-many entity mapping is shown in Figure V-20 on the next page.

FIG. V-20: FORMATION OF VIRTUAL ENTITIES: ONE-TO-MANY ENTITY MAPPINGS



Many-to-one Entity Mappings

One type of mapping exists between virtual and real entities that did not require a separate description in the mappings between real entities. This is the many-to-one mapping, when many virtual entities are created from one real entity. In the case of mappings between real entities, the many-to- one mapping can be represented as a one-to-many mapping going in the opposite direction, since both sides of the mapping are at the operational level. In a many-to-one mapping to the virtual level, two different types of entities are involved, virtual and real. The mapping from real to virtual entities can be expressed using a combination of one-to-one projection mappings from the operational to the virtual level:

(n:1 MAPPINGS) (d) The virtual entities are formed as projections of a real entity:

$$VE_i \equiv \prod_{y=1}^{n} X_y \quad \text{where } X_y = A_{pqr}.$$

$$VA^k{}_{ij} \equiv f(\text{keys of } X_{y,\ y=1,n}).$$

TRANSLATION FUNCTION: $VA^k{}_{ij} = t(\text{keys of } X_{y,y=1,n})$.

AND

$$VE_l \equiv \prod_{y=1}^{n} X_y \quad \text{where } X_y = A_{pqr}.$$

$$VA^k{}_{lm} \equiv f(\text{keys of } X_{y,\ y=1,n}).$$

TRANSLATION FUNCTION: $VA^k{}_{lm} = t(\text{keys of } X_{y,y=1,n})$.

The keys of the resulting virtual entities will be functions of the keys of the real entities from which they are created.

There are no many-to-one entity mappings in the example databases, but the concept is similar to the one-to-many mapping already illustrated.

Entity to Relationship Attribute Mappings

(1:1 MAPPINGS) (e) The virtual entity is formed from one or more attributes of a relationship:

$$VE_i \equiv \{A^r{}_{xyz},\ z=1,n\}.$$

The key of the resulting virtual entity will be a function of the relationship attribute; in turn, the relationship attribute is a function of the keys of the entities involved in the relationship. Assume only one relationship attribute is involved in the mapping, and that it is a function of two entity keys.

$$\therefore\ VA^k{}_{ij} \equiv A^r{}_{xyz}.$$

$$\text{TRANSLATION FUNCTION: } VA^k{}_{ij} = t(A^r{}_{xyz}).$$

The relationship attribute is a function of the keys of the entities involved in the relationship.

$$A^r{}_{xyz} = f(A^k{}_{pqr},\ A^k{}_{abc}).$$

$$\therefore\ VA^k{}_{ij} \equiv f(A^k{}_{pqr},\ A^k{}_{abc}).$$

There are no entity-to-relationship-attribute mappings in the example databases.

B.3.3.2.2 VIRTUAL ENTITY SETS WHICH DO NOT HAVE REAL ENTITY SET COUNTERPARTS

The need for a virtual entity set that does not have a real entity set counterpart deserves special consideration. These virtual entity sets can be identified from the following mappings performed in Step 1: (1) one-to-one entity attribute mappings, (2) one-to-one relationship attribute mappings, or (3) one-to-one relationship attribute to entity attribute mappings. If the same physical object appears as an attribute of more than one database component, it may play an important role in the problem domain and deserve consideration as a virtual entity. In this case, the virtual entity will be formed from either an entity attribute or a relationship attribute using the notation specified above.

For instance, in this problem domain, the owner of a business enterprise is of primary interest because he/she will be the target for any responsible party inquiries. This situation led to the creation of the OWNER entity from the MAN_OWNER attribute of the MANUFACTURER relation, the SHIP_CO_OWNER attribute of the SHIPPING_COMPANY relation, and the WDISPOSER_OWNER attribute of the WASTE_DISPOSER relation, as shown below.

OWNER (VIRTUAL) ≡ MAN_OWNER	$VE_5 \equiv A_{112}$
OWNER (VIRTUAL) ≡ SHIP_CO_OWNER	$VE_5 \equiv A_{212}$
OWNER (VIRTUAL) ≡ WDISPOSER_OWNER	$VE_5 \equiv A_{312}$

The OWNER entity will have four attributes: OWNER_ID, which will be derived from the set of attributes MAN_OWNER, SHIP_CO_OWNER, and WDISPOSER_OWNER. If these three attributes have different domains, a single domain must be selected and values for the other attributes converted to the selected domain. The other three attributes will be MANUF_ID, SHIP_CO_ID, and WDISPOSER_ID, which will be initialized as appropriate.

The virtual entity which has no real entity set counterpart and the mappings from the entity attributes are shown in Figure V-21 on the next page. On the following page, Figure V-22 illustrates all of the virtual and real entities and all mappings from the operational to the virtual level.

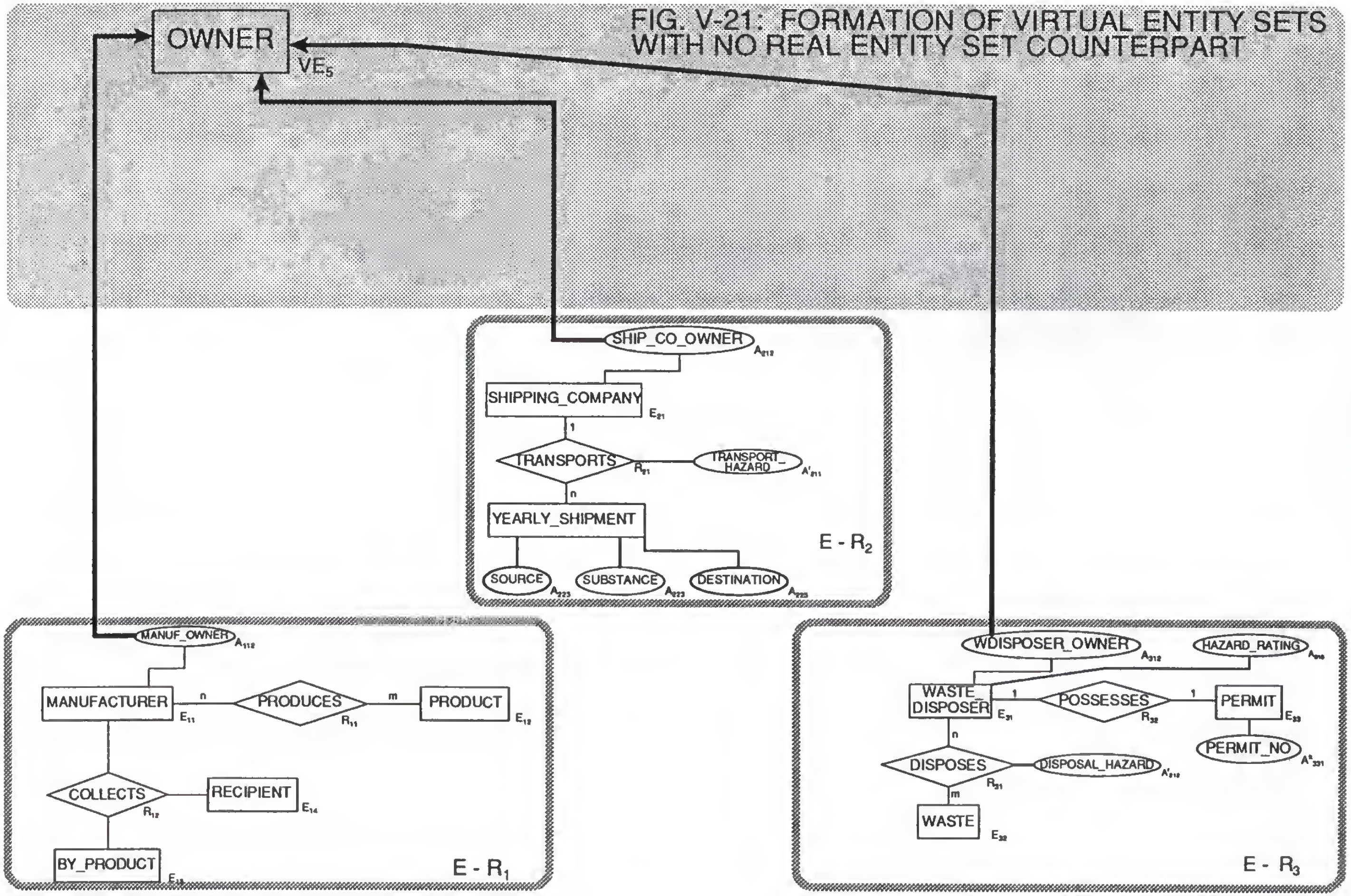


FIG. V-21: FORMATION OF VIRTUAL ENTITY SETS WITH NO REAL ENTITY SET COUNTERPART

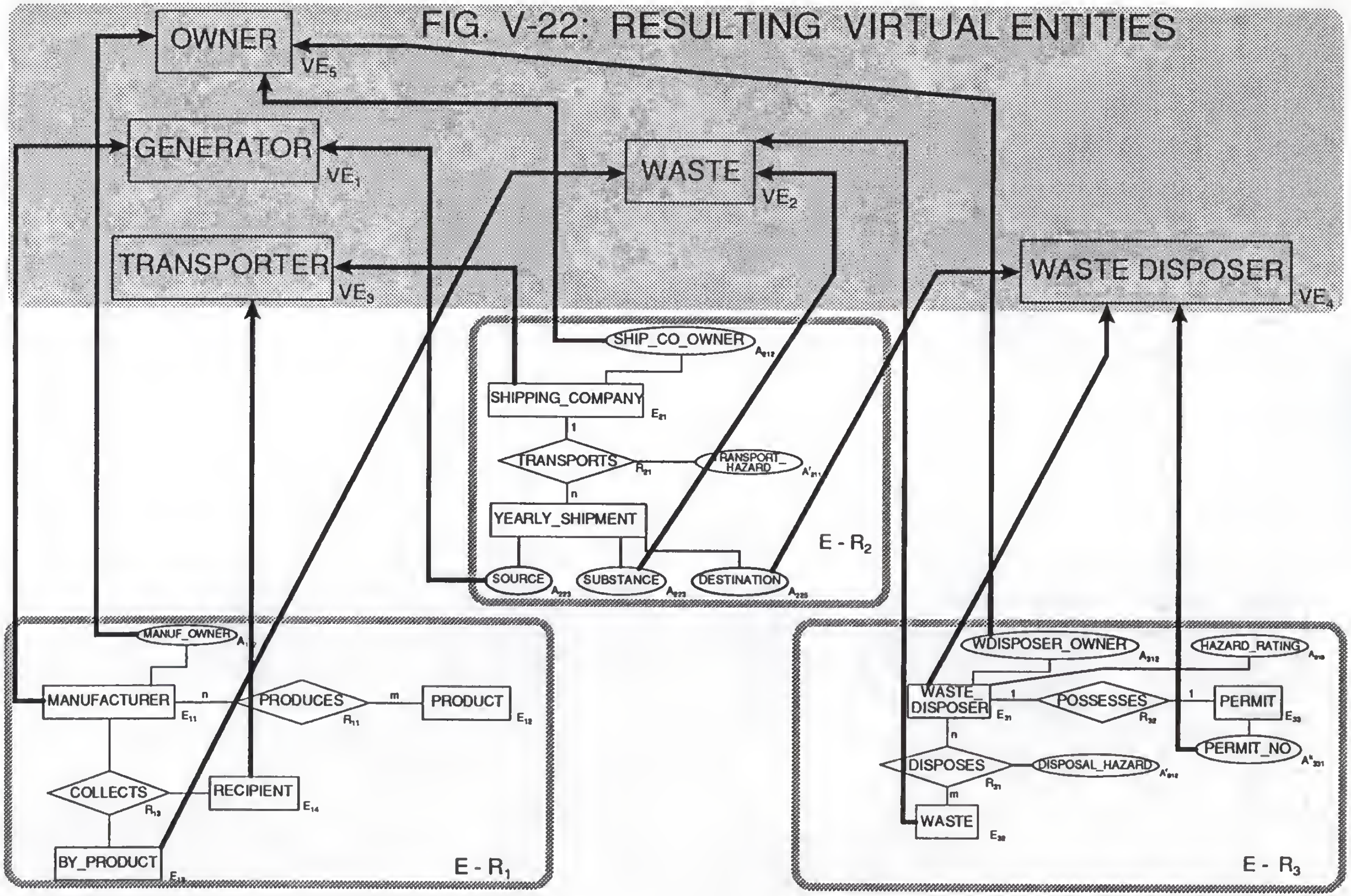


FIG. V-22: RESULTING VIRTUAL ENTITIES

B.3.4 (STEP 4) IDENTIFY RELATIONSHIPS BETWEEN VIRTUAL ENTITIES

Most virtual relationships will be derived from the real relationships in operational databases or from the relationships across database boundaries specified in Step 2.

In forming virtual relationships, we must consider each virtual entity pair and answer the following question:

> How are the virtual entities mapped from real entities?
>
> If both virtual entity sets are created from one-to-one equivalence mappings from real entity sets, any real relationship between the two virtual entities (either R_{ij} or RR_i) is eligible to become a virtual relationship.
>
> Otherwise, each relationship between the real entities involved should be examined to determine whether it holds (in a semantic sense) for the newly created virtual entities.

B.3.4.1 An Example of the Analysis for Determining Virtual Relationships

Examine each virtual entity pair and the relationships in which the entities participate.

VIRTUAL ENTITY PAIR	REAL RELATIONSHIPS	RELATIONSHIPS ACROSS BOUNDARIES
GENERATOR, WASTE	COLLECTS	COLLECTS_T2
GENERATOR, TRANSPORTER	COLLECTS	COLLECTS_T
GENERATOR, WASTE_DISPOSER	none	none
GENERATOR, OWNER	none	none
WASTE, TRANSPORTER	COLLECTS	COLLECTS_T COLLECTS_T2 DELIVERS
WASTE, WASTE_DISPOSER	DISPOSES	DISPOSES_T DELIVERS
WASTE, OWNER	none	none
TRANSPORTER, WASTE_DISPOSER	none	DELIVERS
TRANSPORTER, OWNER	none	none
WASTE_DISPOSER, OWNER	none	none

TABLE V-13: VIRTUAL ENTITY PAIRS AND RELATIONSHIPS IN WHICH THEY PARTICIPATE

As a result of this analysis, we can see that there are three potential virtual relationships that can be mapped directly from real relationships or relationships that cross database boundaries: COLLECTS, DISPOSES, and DELIVERS. Note that, because the decision was made to not include YEARLY_SHIPMENTS as an entity at the virtual level, none of the relationships involving that entity are eligible.

We assume that the information contributed by the DISPOSES relationship is contained within the DELIVERS relation, since all WASTE disposed at a WASTE_DISPOSER facility will have been delivered to the facility. Therefore, we will map only two real relationships to the virtual level: COLLECTS and DELIVERS.

B.3.4.2 EXAMPLES AND DISCUSSION OF RELATIONSHIPS BETWEEN VIRTUAL ENTITIES

A <u>virtual relationship</u> VR_i is described by a subscript i which tells us which relationship it is within the virtual E-R. The virtual relation VR_i is described by the virtual keys of the entities being related and a optional set of relationship attributes. A procedure for developing tuples to be included in the relationship must be specified.

VR_i ({vk_1, ...vk_n} | n≥2; {VRA}).

where $vk_x \varepsilon \{VA^k_{pq}, p=1,e; q=1,f\}$;
p indicates the virtual entity participating in the relationship;
e indicates the number of entities in virtual E-R;
q indicates the key attribute of the virtual entity p;
f indicates the number of key attributes of virtual entity p;
{VRA} is the optional set of relationship attributes and
{VRA} = {} |
$\{VA^r_{i1}\}$ |
$\{a_1, \ldots a_m\}$ | m≥2 and $a_y \varepsilon \{VA^r_{ij}, j=1,m\}$).

PROCEDURE: $VR_i \equiv R_{ij}$,
$VR_i \equiv RR_i$,
OR OTHER SPECIFICATION.

Example from the problem databases:
VR_1 ({VA^k_{11}; VA^k_{21}; VA_{k31}}; {VA^k_{11}})
COLLECTS ({GENERATOR_ID;WASTE_NO;TRANSPORTER_ID};{QTY_COLLECTED})
PROCEDURE: $VR_1 \equiv R_{12}$ COLLECTS (VIRTUAL) ≡ COLLECTS

This notation indicates that the virtual entities VE_1, GENERATOR, VE_2, WASTE, and VE_3, TRANSPORTER, are related, with the inclusion of the key attributes for these virtual entities. As with real relationships, the keys are first presented, separated by a semicolon because of the possibility that more than one attribute may comprise the key of a virtual entity relation. The set of keys is enclosed in brackets and followed by a semicolon to separate the list of key attributes from the list of relationship attributes. Next the relationship attributes are given; in this case, a QTY_COLLECTED attribute is listed.

The notation indicating that this virtual relationship is equivalent to the real relationship COLLECTS means that tuples for the COLLECTS relationship can be used to initialize the virtual relationship tuples.

Example from the problem databases:
VR_2 ({VA^k_{21}; VA_{k31}; VA^k_{41}}; {VA^r_{21}})
DELIVERS ({WASTE_NO; TRANSPORTER_ID; WDISPOSER_ID};
{QTY_DELIVERED})
PROCEDURE: $VR_2 \equiv RR_9$ DELIVERS (VIRTUAL) ≡ DELIVERS

This notation indicates that the virtual entities VE_2, WASTE, VE_3, TRANSPORTER, and VE_4, WASTE DISPOSER, are related, with the inclusion of the key attributes for these virtual entities. As with real relationships, the keys are first presented, separated by a semicolon because of the possibility that more than one attribute may comprise the key of a virtual entity relation. The set of keys is enclosed in brackets and followed by a semicolon to separate the list of key attributes from the list of relationship attributes. Next the relationship attributes are given; in this case, a QTY_DELIVERED attribute is listed.

The notation indicating that this virtual relationship is equivalent to the real relationship between entities in operational databases, DELIVERS, indicates that tuples for this virtual relationship will be derived in the same manner as tuples for the DELIVERS relationship.

Finally, a new relationship must be introduced to associate the new OWNER virtual entity sets with other entity sets which have been mapped from real entity sets. The DBA must specify how to derive tuples for this new relationship.

VR_3 = (($VA^k{}_5$; $VA^k{}_1$; $VA^k{}_3$; $VA^k{}_4$);)
OWNS = ({OWNER_ID; GENERATOR_ID;TRANSPORTER_ID;WDISPOSER_ID});{})

PROCEDURE:

For each MANUFACTURER, SHIPPING_COMPANY, RECIPIENT, or WASTE_DISPOSER in an operational database, create an OWNER tuple, converting the owner attribute to the selected format for the OWNER_ID. For OWNER_IDs which appear in more than one database, combine tuples to reflect that one OWNER has more than one waste handler in the disposal chain.

Adding these three virtual relationships to the virtual E-R diagram results in the final Virtual E-R diagram, which is shown along with the three operational databases from which is it was derived in Figure V-23 on the next page.

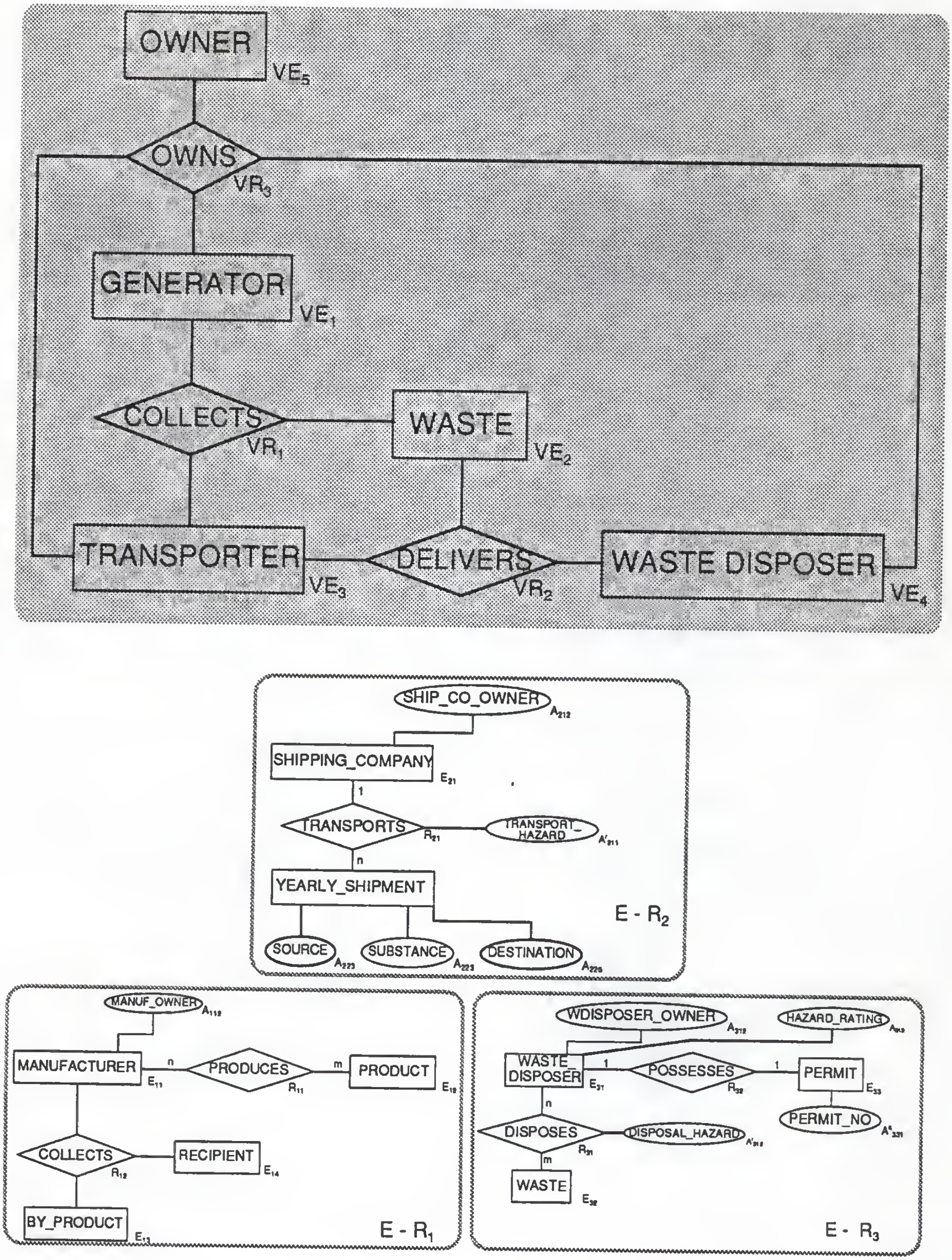


FIG. V-23: OPERATIONAL & VIRTUAL DATABASES

The complete virtual entity-relationship diagram with all attributes is shown in Figure V-24 below.

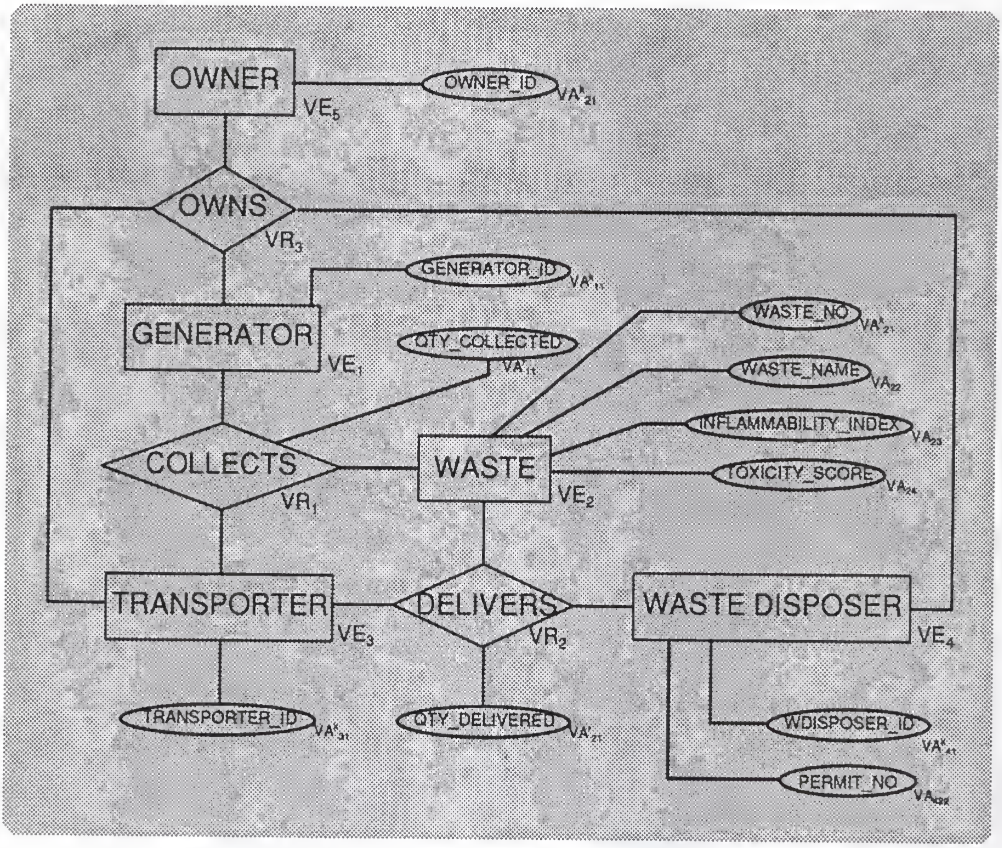


FIG. V-24: FINAL VIRTUAL ENTITY-RELATIONSHIP MODEL

The mappings from the operational level to the virtual level will enable the DSS to translate user queries based on the virtual view of the data into database retrievals using the actual operational database structures. Mappings for virtual entities and relationships that do not have exact counterparts in the operational databases will be especially critical. For instance, in our example virtual E-R, the DELIVERS relationship between TRANSPORTER, WASTE and WASTE_DISPOSER does not have an operational database counterpart. However, we specified a procedure for determining which tuples are involved in the DELIVERS relationship; this procedure involved a join of tuples from the RECEIVES and CONTAINS2 relationships. In turn, the tuples for RECEIVES and CONTAINS2 were derived as a result of the mappings at the operational level. These two relationships were not included as virtual relationships because they involved the operational YEARLY_SHIPMENTS entity, which is not of interest for the problem domain. However, they are part of the DSS background processing for developing the tuples involved in the DELIVERS relationship.

B.4 The Final Virtual Entity-Relationship Model

The final virtual entity-relationship model is the culmination of the steps we have followed in this chapter. It represents the user's view of the information contained in the databases for the problem domain. It does not incorporate every entity, relationship and attribute contained in the operational databases, because all of this information is not critical for the problem domain and users for which the DSS was designed. In addition, the virtual E-R contains new relationships and entities that do not have exact counterparts in the operational databases. These new relationships and entities will be derived from information contained in the operational databases, but they will represent a different arrangement of that information.

CHAPTER VI. SUMMARY OF NOTATION

Throughout Chapter V, notation has been introduced to explain the steps for developing the virtual Entity-Relationship model. This chapter provides, for the reader's convenience, a summary of the notation introduced.

NOTATION FOR THE GIVEN REAL ENTITIES AND RELATIONSHIPS

The following notation is used to describe entities and relationships in the operational databases. First the notation for real entities is given.

Real entity E_{ij},
where i indicates the operational database, or E-R;
j indicates the entity within the operational E-R.

Entity Attributes A_{ijk},
where i indicates the operational database, or E-R;
j indicates the entity within the operational E-R;
k indicates the attribute of the entity.

Key attribute A^k_{ijl},
where superscript k indicates that the attribute is key for an entity;
i indicates the operational database, or E-R;
j indicates the entity within the operational E-R;
l indicates the attribute of the entity.

Relationship between non-key and key attributes $A_{ijk} = f(A^k_{ijl})$.

The following notation is used to describe relationships between entities in a single operational databae.

Real Relationship R_{ij},
where i indicates the operational database, or E-R;
j indicates the relationship within the operational E-R.

Relationship Attributes A^r_{ijk},
where superscript r indicates that the attribute is a relationship attribute;
i indicates the operational database, or E-R;
j indicates the relationship within the operational E-R;
k indicates the attribute of the relationship.

General format for relationship:
R_{ij} $(\{k_1, \ldots k_n\} \mid n \geq 2; \{RA\})$.

where n indicates the number of entities that are related;

$k_x \varepsilon \{A^k_{ipq}, p=1,e; q=1,f\}$; all keys of the related entities are listed;

i indicates the operational database, or E-R; all entities in a real relationship are from the same operational database;

p indicates the entity participating in the relationship;

e indicates the number of entities in operational database i;

q indicates the key attribute of the entity p;

f indicates the number of key attributes of entity p;

{RA} is the optional set of relationship attributes and

$\{RA\} = \{\} \mid$
$\{A^r_{ij1}\} \mid$
$\{a_1, \ldots a_m\} \mid m \geq 2 \text{ and } a_y \varepsilon \{A^r_{ijk}, k=1,m\}\}$.

Association between relationship attributes and entity keys:
$\mathbf{A^r_{ijk} = f(\{k_1, \ldots k_n\})}$ where definitions are same as above.

The assumption made for this paper is that the real entities and relationships in the operational databases will be given. This information will be used by the DBA in specifying the virtual E-R model. The first step in specifying the virtual E-R is to identify mappings at the operational level.

(STEP 1) IDENTIFY MAPPINGS AT THE OPERATIONAL LEVEL

Mappings at the operational level include mappings between real entity sets and mappings involving relationship attributes. These mappings are used to identify equivalences between objects described in different databases.

MAPPINGS BETWEEN REAL ENTITY SETS

Mappings between real entity sets are used to identify real entities that can be mapped directly to the virtual level. Four types of mappings between real entity sets are possible: (1) one-to-one entity equivalence mappings, (2) one-to-one entity projection mappings, (3) one-to-many entity mappings, (4) partial entity mappings. The notation for each of these mapping types is given below.

One-to-one entity equivalence mapping

$$E_{ij} \equiv E_{pq}.$$

$$\therefore \quad A^k{}_{ijk} \equiv A^k{}_{pqr}.$$

TRANSLATION FUNCTION: $A^k{}_{ijk} = t(A^k{}_{pqr})$

One-to-one entity projection mapping

All attributes included in the projection are from the same real entity, E_{pq}.

$$E_{ij} \equiv \{A_{pqr},\ r=1,n\}.$$

Assume that only one attribute is involved in the projection.

$$\therefore \quad A^k{}_{ijk} \equiv A_{pqr}.$$

TRANSLATION FUNCTION: $A^k{}_{ijk} = t(A_{pqr})$.

$$A_{pqr} = f(A^k{}_{pqs}).$$

$$\therefore \quad A^k{}_{ijk} \equiv f(A^k{}_{pqs}).$$

One-to-many entity mappings

$$E_{ij} \equiv \prod_{y=1}^{n} X_y \quad \text{where } X_y = E_{ab} \mid A_{pqr} \text{ and } n \geq 2.$$

$A^k{}_{ijk} \equiv f(\text{keys of } X_y,\ y=1,n,\ n\geq 2)$.

TRANSLATION FUNCTION: $A^k{}_{ijk} = t(\text{keys of } X_y, y=1,n)$.

Partial entity mappings

All attributes on one side of the equivalence sign must be from the same real entity.

$$\{\mathbf{A_{ijk}},\ \mathbf{k=1,m}\} \equiv \{\mathbf{A_{pqr}},\ \mathbf{r=1,m}\}.$$

Assume that only one attribute is involved in the mapping.

TRANSLATION FUNCTION: $A_{ijk} = t(A_{pqr})$.

for all A_{ijk}, $k=1,m$: $A_{ijk} = f(A^k_{ijl})$ AND

for all A_{pqr}, $r=1,m$: $A_{pqr} = g(A^k_{pqs})$.

$$\therefore f(A^k_{ijl}) \equiv g(A^k_{pqs}).$$

MAPPINGS INVOLVING RELATIONSHIP ATTRIBUTES

Mappings involving relationship attributes are used to identify additional information about the physical objects described in the databases. These mappings identify potential virtual entities that are not otherwise represented as entities. Three types of mappings involving relationship attributes are possible: (1) relationship attribute to relationship attribute mappings; (2) relationship attribute to entity mappings; and (3) relationship attribute to entity attribute mappings.

Relationship Attribute to Relationship Attribute Mapping

All attributes on one side of the equivalence sign must be from the same real relationship.

$$\{\mathbf{A^r_{ijk}},\ \mathbf{k=1,m}\} \equiv \{\mathbf{A_{rxyz}},\ \mathbf{z=1,n}\}.$$

Assume that only one relationship attribute is involved in the mapping and that relationships R_{ij} and R_{xy} involve two entities each.

TRANSLATION FUNCTION: $A^r_{ijk} = t(A^r_{xyz})$

$A^r_{ijk} = f(A^k_{pqr}, A^k_{abc})$ and

$A^r_{xyz} = g(A^k_{def}, A^k_{ghi})$.

$$\therefore f(A^k_{pqr}, A^k_{abc}) \equiv g(A^k_{def}, A^k_{ghi}).$$

Relationship Attribute to Entity Mapping

All attributes on the left side of the equivalence sign must be from the same real relationship, R_{ij}.

$$\{\mathbf{A^r_{ijk}},\ \mathbf{k=1,n}\} \equiv \mathbf{E_{xy}}.$$

Assume that only one relationship attribute is involved in the mapping and that relationship R_{ij} involves only two entities.

$\therefore A^{r}{}_{ijk} \equiv A^{k}{}_{xyz}.$

TRANSLATION FUNCTION: $A^{r}{}_{ijk} = t(A^{k}{}_{xyz}).$

$A^{r}{}_{ijk} = f(A^{k}{}_{pqr}, A^{k}{}_{abc}).$

$\therefore f(A^{k}{}_{pqr}, A^{k}{}_{abc}) \equiv A^{k}{}_{xyz}.$

Relationship Attribute to Entity Attribute Mapping

All attributes on one side of the equivalence sign must be from the same real entity or relationship.

$$\{A^{r}ijk,\ k=1,m\} \equiv \{Apqr,\ r=1,n\}.$$

Assume that only one relationship attribute is involved in the mapping, only one entity attribute is involved in the mapping, and relationship R_{ij} involves only two entities.

TRANSLATION FUNCTION: $A^{r}{}_{ijk} = t(A_{pqr}).$

$\therefore A^{r}{}_{ijk} \equiv g(A^{k}{}_{pqs}).$

$A^{r}{}_{ijk} = f(A^{k}{}_{xyz}, A^{k}{}_{abc}).$

$\therefore f(A^{k}{}_{xyz}, A^{k}{}_{abc}) \equiv g(A^{k}{}_{pqs}).$

Once all equivalences are identified in the operational databases, new relationships can be introduced.

(STEP 2) IDENTIFY RELATIONSHIPS BETWEEN REAL ENTITIES IN DIFFERENT OPERATIONAL DATABASES

Relationships between real entities in different operational databases are introduced as a result of equivalences between data items in the different databases. Because these relationships do not exist within a single database, a procedure must be specified for developing tuples for these relationships.

Relationship between real entities in different operational databases

RR_i,

where i indicates which relationship between real entities in different operational databases is being referenced.

Relationship Attributes A^{rr}_{ij},

where superscripts rr indicate that the relationship attribute belongs to a relationship between real entities in different operational databases;

i indicates the relationship;

j indicates the attribute of the relationship.

General format for relationship:

RR_i ({k_1, ...k_n} | n≥2; {RRA}).

where k_x ε {$A^k{}_{pqr}$, p=1,d; q=1,e; r=1,f};

p indicates the operational database, or E-R;

d indicates the number of operational databases;

q indicates the entity participating in the relationship;

e indicates the number of entities in operational database p;

r indicates the key attribute of the entity q;

f indicates the number of key attributes of entity q;

{RRA} is the optional set of relationship attributes and

{RRA} = {} |

{$A^{rr}{}_{i1}$} |

{a_1, ...a_m} | m≥2 and a_y ε {$A^{rr}{}_{ij}$, j=1,m}).

PROCEDURE: $RR_i \equiv R_{ij}$ OR OTHER SPECIFICATION.

A <u>transferred relationship</u> will assume the same relationship name as the original relationship, with a _T appended. Relationship attribute names will also have _T appended. If a relationship is transferred more than once, each additional transferred relationship will be sequentially numbered, with the number appended after the _T.

Association between relationship attributes and entity keys:

$$A^{rr}_{ij} = f(\{k1, \ldots kn\})$$

where definitions are same as above.

Once all mappings and relationships at the operational level are identified, mappings to the virtual level can be developed.

(STEP 3) IDENTIFY MAPPINGS FROM REAL TO VIRTUAL ENTITY SETS

Virtual entities may be created from real entity sets or without real entity set counterparts.

One-to-one Entity Equivalence Mappings

$$VE_i \equiv E_{pq}.$$

$$\therefore \quad VA^k{}_{ij} \equiv A^k{}_{pqr}.$$

$$\text{TRANSLATION FUNCTION:} \quad VA^k{}_{ij} = t(A^k{}_{pqr}).$$

One-to-one entity projection mapping

All attributes included in the projection are from the same real entity.

$$VE_i \equiv \{A_{pqr},\ r=1,n\}.$$

Assume that only one attribute is involved in the projection.

$$\therefore \quad VA^k{}_{ij} \equiv A_{pqr}.$$

$$\text{TRANSLATION FUNCTION:} \quad VA^k{}_{ij} = t(A_{pqr}).$$

$$A_{pqr} = f(A^k{}_{pqs}).$$

$$\therefore \quad VA^k{}_{ij} \equiv f(A^k{}_{pqs}).$$

One-to-many entity mappings

$$VE_i \equiv \prod_{y=1}^{n} X_y \quad \text{where } X_y = E_{ab} \mid A_{pqr}. \text{ and } n \geq 2.$$

$$VA^k{}_{ij} \equiv f(\text{keys of } X_y,\ y=1,n).$$

$$\text{TRANSLATION FUNCTION: } VA^k{}_{ij} = t(\text{keys of } X_{y,y=1,n}).$$

Many-to-one Entity Mappings

$$VE_i \equiv \prod_{y=1}^{n} X_y \quad \text{where } X_y = A_{pqr}.$$

$$VA^k{}_{ij} \equiv f(\text{keys of } X_y,\ y=1,n).$$

$$\text{TRANSLATION FUNCTION: } VA^k{}_{ij} = t(\text{keys of } X_{y,y=1,n}).$$

AND

$$VE_1 \equiv \prod_{y=1}^{n} X_y \quad \text{where } X_y = A_{pqr}.$$

$$VA^k_{lm} \equiv f(\text{keys of } X_{y,}\ y=1,n).$$

TRANSLATION FUNCTION: $VA^k_{lm} = t(\text{keys of } X_{y,y=1,n}).$

Entity to Relationship Attribute Mappings

All relationship attributes must be from the same real relationship.

$$\mathbf{VE_i \equiv \{A^r_{xyz},\ z=1,n\}.}$$

Assume only one attribute is involved in the mapping, and it is a function of two entity keys.

$$\therefore VA^k_{ij} \equiv A^r_{xyz}.$$

TRANSLATION FUNCTION: $VA^k_{ij} = t(A^r_{xyz}).$

$$A^r_{xyz} = f(A^k_{pqr},\ A^k_{abc}).$$

$$\therefore VA^k_{ij} \equiv f(A^k_{pqr},\ A^k_{abc}).$$

Attribute mappings for virtual entities

$$\mathbf{VA_{ij} \equiv A_{pqr}.}$$

TRANSLATION FUNCTION: $VA_{ij} = t(A_{pqr}).$

(STEP 4) IDENTIFY RELATIONSHIPS BETWEEN VIRTUAL ENTITIES

Relationships between virtual entities will be specified using the following notation.

Virtual Entity VE_i,
where i indicates the entity within the virtual E-R.

Virtual Entity Attributes VA_{ij},
where i indicates the entity within the virtual E-R;
j indicates the attribute of the virtual entity.

Key attribute VA^k_{il},
where superscript k indicates that the attribute is key for an entity;
i indicates the entity within the virtual E-R;
l indicates the attribute of the virtual entity that is key.

Relationship between non-key and key attributes: $VA_{ij} = f(VA^k_{il})$.

Virtual Relationships VR_i,
where i indicates the relationship within the virtual E-R.

Virtual Relationship Attributes VA^r_{ij},
where superscript r indicates that the attribute is a relationship attribute;
i indicates the relationship within the virtual E-R;
j indicates the attribute of the virtual relationship.

General format for relationship:
VR_i $(\{vk_1, \ldots vk_n\} \mid n \geq 2; \{VRA\})$.

where $vk_x \in \{VA^k_{pq}, p=1,e; q=1,f\}$;
p indicates the virtual entity participating in the relationship;
e indicates the number of entities in virtual E-R;
q indicates the key attribute of the virtual entity p;
f indicates the number of key attributes of virtual entity p;
{VRA} is the optional set of relationship attributes and
$\{VRA\} = \{\} \mid$
$\{VA^r_{il}\} \mid$
$\{a_1, \ldots a_m\} \mid m \geq 2$ and $a_y \in \{VA^r_{ij}, j=1,m\})$.

PROCEDURE: $VR_i \equiv R_{ij}$,
$VR_i \equiv RR_i$,
OR OTHER SPECIFICATION.

Association between virtual relationship attributes and virtual entity keys:

$VA^{r}_{ij} = f(\{vk_1, \ldots vk_n\})$

where definitions are same as above.

CHAPTER VII. CONCLUSIONS AND FUTURE RESEARCH

The final virtual entity-relationship model represents the user's view of the information contained in the databases for the problem domain. It does not incorporate every entity, relationship and attribute contained in the operational databases, because all of this information is not critical for the problem domain and users for which the DSS was designed. The mappings from the operational level to the virtual level will enable the DSS to translate user queries based on the virtual view of the data into database retrievals using the actual operational database structures.

A. PROBLEMS NOT ADDRESSED

Problems of efficiency for implementation have not been addressed. It may be necessary to generate several sets of relationship tuples in order to satisfy a single user query for the example virtual E-R; combined with performing retrievals from multiple heterogeneous databases and reconciling information in different formats, much overhead processing will be necessary.

A problem that has not been addressed is the existence of null values in the databases and their impact on the relationships and mappings. Attributes which may assume null values are usually excluded from consideration as key values. Therefore, the mappings which involve entity and relationship attributes must be modified to consider attributes which may assume null values.

It is possible that entity sets may be equivalent only when certain conditions are met or only for selected entities meeting certain criteria; this issue should require only a slight extension of the model presented but has not yet been examined.

Other potential integrity problems have not been addressed, such as the situation when different databases have unequal values for an attribute that should be equal. Introducing the notion of semantic integrity is a challenge left to future researchers.

B. FUTURE RESEARCH

Several other types of mappings between entity sets are possible but have not been addressed in this paper. An example is an entity set in one database that is an aggregate of an entity set in another database. Another example is an entity set in one database that includes only entities from a second entity set which have certain attribute values. Many types of mappings are possible by applying relational algebra manipulations to entity sets; these mappings have been left for further research.

The process of specifying mappings is long and complex, and it requires a great deal of human judgment. An artificially intelligent tool could be designed to aid the DSS DBA in specifying mappings and developing the virtual entity-relationship model. Although such a tool can guide the DBA in developing these specifications, human judgment will be required to establish the semantic equivalences.

BIBLIOGRAPHY

Chen76 Chen, Peter Pin-Shan. "The Entity-Relationship Model--Toward a Unified View of Data," ACM Transactions on Database Systems 1, 1 (March 1976), pp. 9-36.

Date86 Date, C. J. An Introduction to Database Systems. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc. (1986).

Gorr71 Gorry, G. A. and Scott Morton, Michael, "A Framework for Management Information Systems," Sloan Management Review 13, 1, pp. 55-77.

Hawr84 Hawryszkiewycz, I. T. Database Analysis and Design. Chicago: Science Research Associates, Inc. (1984).

Spra83a Sprague, Ralph H. and Hugh J. Watson. "BIT BY BIT: Toward Decision Support Systems," In Decision Support Systems, William C. House, Ed. Petrocelli Books, Inc., New York (1983), pp. 15-31.

Spra83b Sprague, Ralph H. "A Framework for the Development of Decision Support Systems," in Decision Support Systems, William C. House, Ed. Petrocelli Books, Inc., New York (1983), pp. 85-123.

Wats83 Watson, Hugh J. and Marianne M. Hill. "Decision Support Systems or What Didn't Happen with MIS," Interfaces 13, 5 (October 1983), pp. 81-88.

ACHIEVING LOGICAL SEPARATION
OF THE DECISION SUPPORT SYSTEM DATABASE
FROM OPERATIONAL DATABASES

by

D'Ann Naylor Rifai

B.S., Oklahoma State University, 1983
M.B.A., Kansas State University, 1984

AN ABSTRACT OF A THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988

Ralph Sprague, one of the foremost leaders in the field of decision support systems (DSS), has stated that "most successful DSS's have found it necessary to create a DSS database which is logically separate from other operational databases." Logical separation implies that the user's view of the data which the DSS provides is independent of the physical structure of the operational databases supporting the DSS. Achieving this goal requires a sophisticated data management subsystem for the DSS.

Researchers in the area of DSS have not concentrated upon the data management subsystem of the DSS, because database technology is in its mature stages. Few have addressed the problem of identifying and combining data from information sources which describe the same physical objects in different ways. However, in the DSS environment, information is obtained from multiple heterogeneous databases. Therefore, it is possible that an entity from one database can be represented in any of the following ways in a second database: an entity, a relationship, an attribute of another entity, or an attribute of a relationship. This fact presents a problem when combining information from different databases.

This paper addresses the problem by introducing the notion of a mapping, which identifies database components which provide different representations of the same physical object. Mappings enable the DSS to locate all information in different databases about a given physical object, therefore allowing the system to provide a flexible representation of the data to the user. This "virtual" view of the data is independent of the physical structure of the operational databases, because the mappings provide the link between the virtual and the operational levels.