The Entity-Relationship Model—Toward a Unified View of Data

PETER PIN-SHAN CHEN

Massachusetts Institute of Technology

A data model, called the entity-relationship model, is proposed. This model incorporates some of the important semantic information about the real world. A special diagrammatic technique is introduced as a tool for database design. An example of database design and description using the model and the diagrammatic technique is given. Some implications for data integrity, information retrieval, and data manipulation are discussed.

The entity-relationship model can be used as a basis for unification of different views of data: the network model, the relational model, and the entity set model. Semantic ambiguities in these models are analyzed. Possible ways to derive their views of data from the entity-relationship model are presented.

Key Words and Phrases: database design, logical view of data, semantics of data, data models, entity-relationship model, relational model, Data Base Task Group, network model, entity set model, data definition and manipulation, data integrity and consistency
CR Categories: 3.50, 3.70, 4.33, 4.34

1. INTRODUCTION

The logical view of data has been an important issue in recent years. Three major data models have been proposed: the network model [2, 3, 7], the relational model [8], and the entity set model [25]. These models have their own strengths and weaknesses. The network model provides a more natural view of data by separating entities and relationships (to a certain extent), but its capability to achieve data independence has been challenged [8]. The relational model is based on relational theory and can achieve a high degree of data independence, but it may lose some important semantic information about the real world [12, 15, 23]. The entity set model, which is based on set theory, also achieves a high degree of data independence, but its viewing of values such as "3" or "red" may not be natural to some people [25].

This paper presents the entity-relationship model, which has most of the advantages of the above three models. The entity-relationship model adopts the more natural view that the real world consists of entities and relationships. It

A version of this paper was presented at the International Conference on Very Large Data Bases, Framingham, Mass., Sept. 22–24, 1975.

Author's address: Center for Information System Research, Alfred P. Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139.

ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, Pages 9–36.

incorporates some of the important semantic information about the real world (other work in database semantics can be found in [1, 12, 15, 21, 23, and 29]). The model can achieve a high degree of data independence and is based on set theory and relation theory.

The entity-relationship model can be used as a basis for a unified view of data. Most work in the past has emphasized the difference between the network model and the relational model [22]. Recently, several attempts have been made to reduce the differences of the three data models [4, 19, 26, 30, 31]. This paper uses the entity-relationship model as a framework from which the three existing data models may be derived. The reader may view the entity-relationship model as a generalization or extension of existing models.

This paper is organized into three parts (Sections 2–4). Section 2 introduces the entity-relationship model using a framework of multilevel views of data. Section 3 describes the semantic information in the model and its implications for data description and data manipulation. A special diagrammatric technique, the entity-relationship diagram, is introduced as a tool for database design. Section 4 analyzes the network model, the relational model, and the entity set model, and describes how they may be derived from the entity-relationship model.

2. THE ENTITY-RELATIONSHIP MODEL

2.1 Multilevel Views of Data

In the study of a data model, we should identify the levels of logical views of data with which the model is concerned. Extending the framework developed in [18, 25], we can identify four levels of views of data (Figure 1):

(1) Information concerning entities and relationships which exist in our minds.

(2) Information structure—organization of information in which entities and relationships are represented by data.

(3) Access-path-independent data structure—the data structures which are not involved with search schemes, indexing schemes, etc.

(4) Access-path-dependent data structure.

In the following sections, we shall develop the entity-relationship model step by step for the first two levels. As we shall see later in the paper, the network model, as currently implemented, is mainly concerned with level 4; the relational model is mainly concerned with levels 3 and 2; the entity set model is mainly concerned with levels 1 and 2.

2.2 Information Concerning Entities and Relationships (Level 1)

At this level we consider entities and relationships. An *entity* is a "thing" which can be distinctly identified. A specific person, company, or event is an example of an entity. A *relationship* is an association among entities. For instance, "father-son" is a relationship between two "person" entities.[1]

[1] It is possible 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 [27]. He should define what are entities and what are relationships so that the distinction is suitable for his environment.

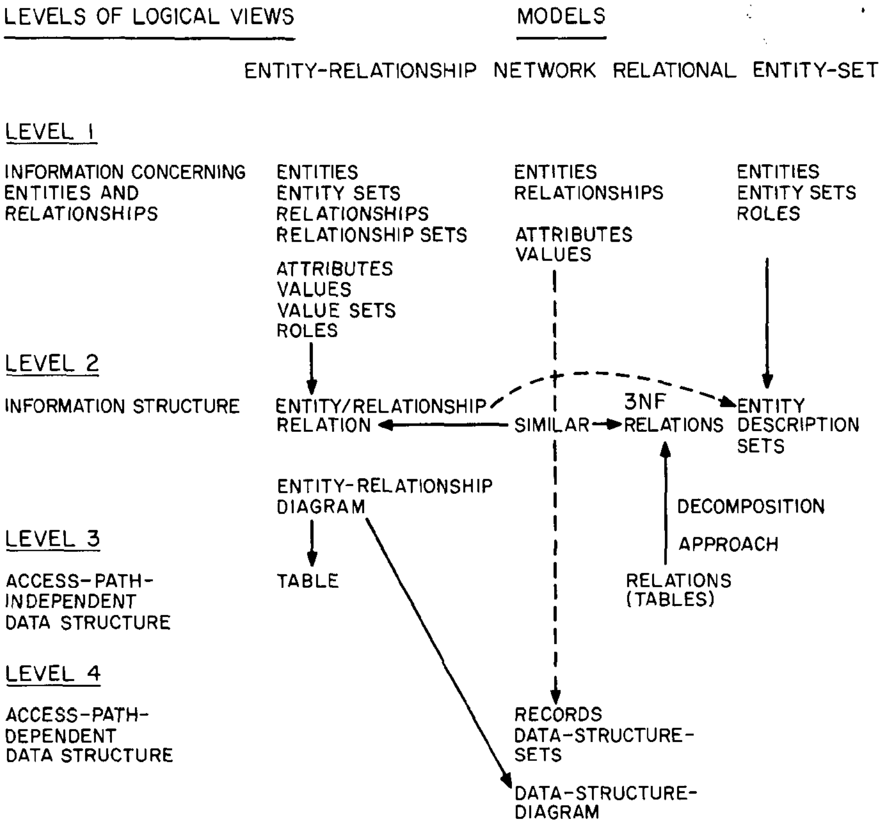

Fig. 1. Analysis of data models using multiple levels of logical views

The database of an enterprise contains relevant information concerning entities and relationships in which the enterprise is interested. A complete description of an entity or relationship may not be recorded in the database of an enterprise. It is impossible (and, perhaps, unnecessary) to record every potentially available piece of information about entities and relationships. From now on, we shall consider only the entities and relationships (and the information concerning them) which are to enter into the design of a database.

2.2.1 Entity and Entity Set. Let e denote an entity which exists in our minds. Entities are classified into different *entity sets* such as EMPLOYEE, PROJECT, and DEPARTMENT. There is a predicate associated with each entity set to test whether an entity belongs to it. For example, if we know an entity is in the entity set EMPLOYEE, then we know that it has the properties common to the other entities in the entity set EMPLOYEE. Among these properties is the aforementioned test predicate. Let E_i denote entity sets. Note that entity sets may not be mutually disjoint. For example, an entity which belongs to the entity set MALE-PERSON also belongs to the entity set PERSON. In this case, MALE-PERSON is a subset of PERSON.

2.2.2 Relationship, Role, and Relationship Set. Consider associations among entities. A *relationship set*, R_i, is a mathematical relation [5] among n entities, each taken from an entity set:

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

and each tuple of entities, $[e_1, e_2, \ldots, e_n]$, is a *relationship*. Note that the E_i in the above definition may not be distinct. For example, a "marriage" is a relationship between two entities in the entity set PERSON.

The *role* of an entity in a relationship is the function that it performs in the relationship. "Husband" and "wife" are roles. The ordering of entities in the definition of relationship (note that square brackets were used) can be dropped if roles of entities in the relationship are explicitly stated as follows: $(r_1/e_1, r_2/e_2, \ldots, r_n/e_n)$, where r_i is the role of e_i in the relationship.

2.2.3 Attribute, Value, and Value Set. 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", "Peter", and "Johnson" are values. Values are classified into different *value sets*, such as FEET, COLOR, FIRST–NAME, and LAST–NAME. There is a predicate associated with each value set to test whether a value belongs to it. A value in a value set may be equivalent to another value in a different value set. For example, "12" in value set INCH is equivalent to "1" in value set FEET.

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:

$$f\colon E_i \text{ or } R_i \rightarrow V_i \text{ or } V_{i_1} \times V_{i_2} \times \cdots \times V_{i_n}.$$

Figure 2 illustrates some attributes defined on entity set PERSON. The attribute AGE maps into value set NO–OF–YEARS. An attribute can map into a Cartesian product of value sets. For example, the attribute NAME maps into value sets FIRST–NAME, and LAST–NAME. Note that more than one attribute may map from the same entity set into the same value set (or same group of value sets). For example, NAME and ALTERNATIVE–NAME map from the entity set EMPLOYEE into value sets FIRST–NAME and LAST–NAME. Therefore, attribute and value set are different concepts although they may have the same name in some cases (for example, EMPLOYEE–NO maps from EMPLOYEE to value set EMPLOYEE–NO). This distinction is not clear in the network model and in many existing data management systems. Also note that an attribute is defined as a function. Therefore, it maps a given entity to a single value (or a single tuple of values in the case of a Cartesian product of value sets).

Note that relationships also have attributes. Consider the relationship set PROJECT–WORKER (Figure 3). The attribute PERCENTAGE–OF–TIME, which is the portion of time a particular employee is committed to a particular project, is an attribute defined on the relationship set PROJECT–WORKER. It is neither an attribute of EMPLOYEE nor an attribute of PROJECT, since its meaning depends on both the employee and project involved. The concept of attribute of relationship is important in understanding the semantics of data and in determining the functional dependencies among data.

2.2.4 Conceptual Information Structure. We are now concerned with how to organize the information associated with entities and relationships. The method proposed in this paper is to separate the information about entities from the infor-

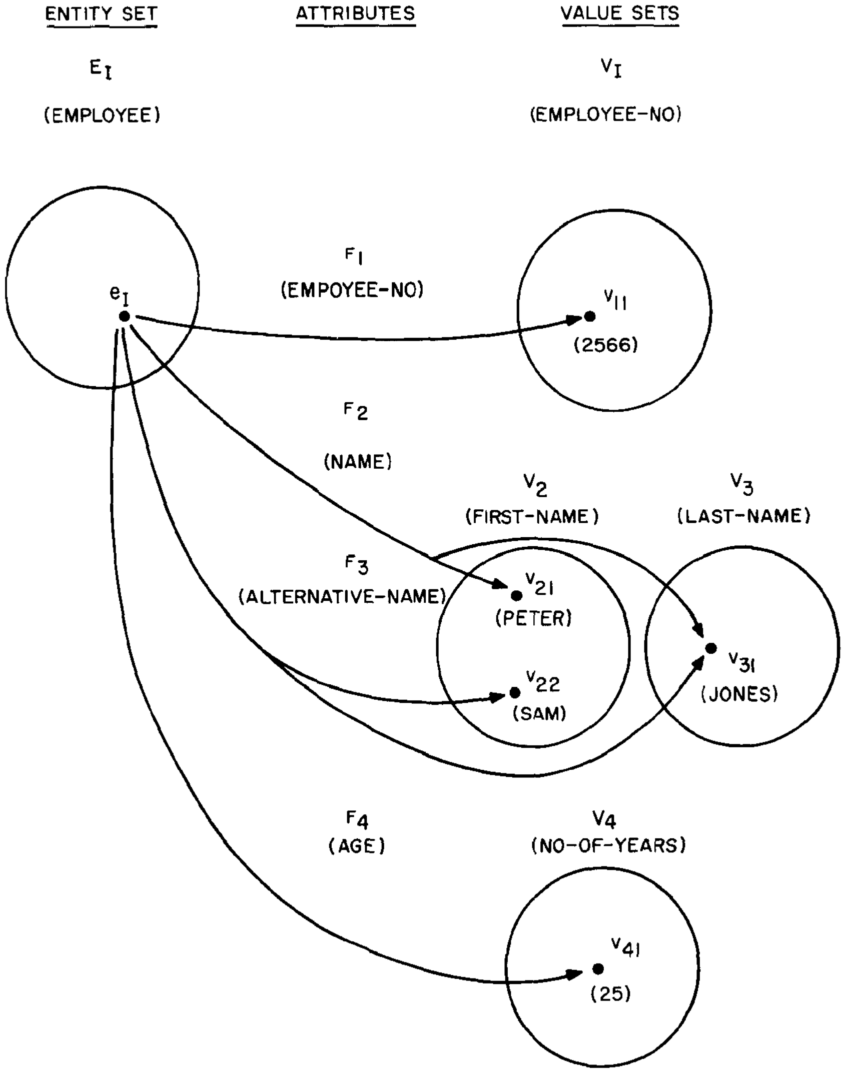

Fig. 2. Attributes defined on the entity set PERSON

mation about relationships. We shall see that this separation is useful in identifying functional dependencies among data.

Figure 4 illustrates in table form the information about entities in an entity set. Each row of values is related to the same entity, and each column is related to a value set which, in turn, is related to an attribute. The ordering of rows and columns is insignificant.

Figure 5 illustrates information about relationships in a relationship set. Note that each row of values is related to a relationship which is indicated by a group of entities, each having a specific role and belonging to a specific entity set.

Note that Figures 4 and 2 (and also Figures 5 and 3) are different forms of the same information. The table form is used for easily relating to the relational model.