Entity-Relationship Approach to
Information Modeling and Analysis, P.P. Chen (ed.)
Elsevier Science Publishers B.V. (North-Holland)

A PRELIMINARY FRAMEWORK FOR ENTITY-RELATIONSHIP MODELS

PETER PIN-SHAN CHEN

Graduate School of Management
University of California, Los Angeles
Los Angeles, California 90024
U.S.A.

ABSTRACT

Many models have been proposed for use in information modeling and analysis. Most of them use the concepts of entity, relationship, and attribute. The major difference of these models are their definitions and use of the concepts of "relationship" and and "attribute". This paper proposes a framework for classifying entity-relationship models and discusses how these models might be translated from one to the other.

I. INTRODUCTION

In recent years, many entity-relationship models have been proposed for use in information modeling and analysis. Most of them are based on the concepts of "entity", "relationship", and "attribute". The differences between these entity-relationship models are their interpretations of (and limitations on) the concepts of "relationship" and "attribute". The supporters of a particular model may claim that their model is more "powerful" than other models, while others may claim their models are more "elegant" since fewer primitive concepts are used. It is very difficult for a newcomer to this field to fully understand these claims and to evaluate the merits of different models. Therefore, a framework for entity-relationship models is needed to put them in perspective.

In this paper, a preliminary framework for entity-relationship models is proposed. The framework is based on the capabilities and limitations of "relationship" and "attribute" in each model. In terms of "relationship", models are diffrentiated by whether they allow n-ary relationships or allow only binary relationships. In terms of "attribute", some models allow "attribute of relationship", while some others allow no attributes at all. By diffrentiating on the treatment of "relationships " and "attributes", we can present a framework for classifying entity-relationship models.

The purpose of this paper is not only to classify entity-relationship models into different categories but also to demonstrate the feasibility of translating from one type of entity-relationship model to another. If such translations are feasible, there are several significant implications for information modeling and analysis. First, the systems analyst may use their favorite model to model the real world and then covert it into another model for presentation to others. Second, it may be possible to implement a database management system which supports several types of entity-relationship models. Third, it may be able to demonstrate that different types of entity-relationship models are equivalent. The equivalence of "data models" has been a subject of research for several researchers (for example, [SIBL74]). We hope that our paper can stimulate more research in this direction.

II. CLASSIFICATION OF ENTITY-RELATIONSHIP MODELS

Entity-relationship models can be divided into two major categories based on the type of relationship allowed in the model. The first category is called Generalized (N-ary) Entity-Relationship Models (GERM), which allows relationships defined on more than two entities. The second category is called Binary Entity-Relationship Models (BERM) which allows at most two entities to be involved in a relationship.

Each of these two major categories can be further divided into three subcategories depending on their treatment of "attributes." The first subcategory of models allows attributes for both entity and relationship. The second subcategory allows attributes for entities only. The third subcategory does not allow attributes at all.

Figure 1 illustrates a framework for entity-relationship models. An example of a GERM which allows attributes for both entities and relationships is the original version of the Entity-Relationship Model [CHEN76, DOSS80, SCHE80]. A graphical representation of the ERM's in this category is given in Figure 2. As shown, a project can participate in "m" (i.e., "many") "WORKS-FOR" relationships, and an employee can participate in "N" (i.e., "many") "WORKS-FOR" relationships. In other words, an employee can work for several projects, and a project can have several employees as workers. Note the three-way relationship between the SUPPLIER, PART, and PROJ entities. The SUPPLIED relationship indicates that a particular supplier supplied a particular part to a particular project. Note in this Entity-Relationship Diagram (ERD) that both entities and relationships can have attributes. For example, a SUPPLY entity has attribute S# and SNAME, and a SUPPLIED relationship has an attribute DATE.

The second subcategory of the GERM's does not allow attributes for relationships. If a "relationship" has an attribute, the analyst will create a "high-level entity" to replace it. Figure 3(a) indicates the graphical representation of this type of ERM. Note that the "SUPPLIED-BY" relationship has been converted into a "high-level entity" called "SUPPLIES", and "DATE" is an attribute of the newly-created SUPPLIES entities. Note that "relationships" are represented by a "straight line" (or arc) between entity types. If the SUPPLIED-BY relationship in Figure 2 does not have any attribute, Figure 3(b) will be the new ERD. Note that the "SUPPLIED-BY" relationship is represented by a "dot", which is the intersection of three lines originating from three entity types: PROJ, PART, and SUPPLIER. The concept of a "high-level entity" and the graphical symbol to represent it have been discussed in [SCHE80, DOSS80, CHEN80]. However, in these papers, the conversion of a "relationship" into a "high-level entity" is done at the analyst's option and is not enforced when the relationship has an attribute.

The third subcategory is a GERM with no attributes at all. In this case, Figure 3(b) will be converted into Figure 4. To the best of our knowledge, no one has proposed an ERM in this subcategory.

The binary ERM's can also be divided into three subcategories, which are designated as (4), (5), and (6) in Figure 1. The (4) subcategory is a binary ERM which allows attributes for both entities and relationships. Figure 5 illustrates an ERD for this kind of ERM. An example of this kind of ERM is the data model used in the IBM DB/DC data dictionary system.

Subcategory (5) in Figure 1 is the class of ERM which allows attributes for entities only. This subcategory can be further divided into two groups depending on whether or not "many-to-many" relationships are allowed. Subcategory (5.1) denotes a class of BERM's which allow many-to-many relationships. Figure 6, which correspond to Figure 2, illustrates an ERD based on this kind of BERM. Note that "WORK-FOR" is a many-to-many relationship. Note also that Figure 6 is the same as Figure 3(a). However, Figure 6 is the only acceptable graphical representation corresponding to Figure 2 for this kind of BERM. Figure 3(b) is not acceptable since it contains a 3-ary relationship "SUPPLIED-BY". Any n-ary relationship has to be converted into an "entity" in a binary model. Therefore, no matter whether the "SUPPLIED-BY" relationship has an attribute or not, Figure 6 is the corresponding ERD for the BERM with attributes for entities only. Examples of this type of BERM are the data models used in many of the commercially available data dictionary systems. Another example is the Simplified ERM used in [FURT81, LUSK81].

Subcategory (5.2) is the class of BERM, which does not allow "many-to-many" relationships. In other words, only "one-to-one" or "one-to-many" relationships are allowed. This subcategory can be further divided into two groups depending on whether the relationship is directional or non-directional. Subcategory (5.2.1) denotes the class of BERM's in which the relationships are non-directional. Figure 7, which corresponds to Figure 6, illustrates an ERD based on this kind of BERM.

Subcategory (5.2.2) denotes the class of BERM's in which the relationships are directional. Each relationship has a direction: it starts with a "parent" entity and ends with a "child" entity. Although it may be possible to access the "parent" entities from a "child" entity, the treatment of these two types of entities is usually quite different. This kind of BERM can be found in many conventional database management systems. Let us divide this subcategory into three groups depending on the number of "parent" entities allowed for a "child" entity.

Subcategory (5.2.2.1) is the class of BERM's in which only one "parent" entity is allowed for a "child" entity and no attributes are allowed for relationships. An example of this kind of BERM is a "pure" hierarchical database management system. Figure 8 illustrates an ERD, which is acceptable in this kind of BERM.

Subcategory (5.2.2.2) is the class of BERM's in which one "physical parent" entity and one "logical parent" entity are allowed for a "child" entity. An example of this kind of BERM is the data model used in the IBM's IMS system. Figure 9 illustrates an ERD which is acceptable in this kind of BERM.

Subcategory (5.2.2.3) is the class of BERM's in which many "parent" entities are allowed for a "child" entity. Figure 10, which is corresponding to Figure 7, illustrates an ERD for this kind of BERM. The CODASYL data model can be classified into this subcategory.

Subcategory (6) is the class of BERM's in which no attributes are allowed. This kind of BERM can be respresented by either Figure 11 or Figure 12. Note that in Figure 12 the entity identifying attribute is used to denote the entity itself. For example, PROJ# is used to denote "PROJ" entity itself. The binary data model [ABRI74] and the Entity set model [SEKO73] can be considered as close to the one represented by Figure 12, while the Functional model [SIBL77] can be considered as close to the one represented by Figure 11 after some modifications.

III. OBSERVATIONS

There are a few observations based on this framework of entity-relationship models:

(a) There is a spectrum of ERM's. On the one end, there is a GERM in which attributes for both entities and relationships are allowed. On the other end, there is a BERM in which no attributes are allowed. The conversion from the former to the latter involves the conversion of n-ary relationships into "entities" and the conversion of attributes into "entities". The reverse conversion is possible, but it needs human inputs to specify what attributes are and what entities are.

(b) Each kind of ERM has its own appeal in certain situations. It is difficult to judge which kind of ERM is always superior than other types.

(c) Although little work has been done for models in category (2), it is conceivable that a theory can be developed along the line of [DOSS80, SCHE80, CHEN80] as discussed earlier in the paper. However, research work on the ERM's in category (3) would be of theoretical value, but its practical value needs to be justified.

IV. SUMMARY

In this paper, we have presented a framework for entity-relationship models. We have also discussed previous research work or commercial systems, which fit into each of these categories. We have tried to use the same example to demonstrate the feasibility of translation from an ERD of one category to an ERD in another category. We hope that the results of this paper can increase the understanding of different types of ERM's and can stimulate more research in the Entity-Relationship approach.

REFERENCES

[ABRI74] Abrial, J.R., Data Semantics, in: Klimbie, J.W. and Koffemann, K.L. (eds.), Data Base Management (North-Holland, Amsterdam, 1974).

[CHEN76] Chen, P.P., The entity relationship model: toward a unified view of data, ACM Transactions on Database Systems, 1, 1 (March 1976),9-37.

[CHEN80] Chen, P.P., Entity-Relationship diagrams and English sentence structures, (abstract only), in: Chen, P.P. (ed.), Entity-Relationship Approach to Systems Analysis and Design (North-Holland, Amsterdam, June 1980).

[DOSS80] dos Santos, C.S., Neuhold, E.J. and Furtado, A.L., A data type approach to the entity-relationship model, in: Chen, P.P. (ed.), Entity- Relationship Approach to Systems Analysis and Design (North-Holland, Amsterdam, 1980).

[FURT83] Furtado, A.L. Veloso, P.A.S., and de Castilho, J.M.V., Verification and testing of S-ER representations, in: Chen, P.P. (ed.), Entity-Relationship to Information Modeling and Analysis (Elsevier Science Publishers B.V., Amsterdam, 1983)

[LUSK83] Lusk E., Petrie G., and Overbeek, R., Item tracking entity-relationship models, in: Chen, P.P. (ed.), Entity-Relationship Approach to Information Modeling and Analysis (Elsevier Science Publishers B.V., Amsterdam, 1983)

[SCHE80] Scheuermann, P., Schiffner, G. and Weber, H., Abstraction Capabilities and invariant properties modeling within the entity-relationship approach, in: Chen, P.P. (ed.), Entity-Relationship Approach to Systems Analysis and Design (North-Holland, Amsterdam, June 1980).

[SENK73] Senko, M.E., Altman, E.B., Astrahan, M.M. and Fehder, P.L., Data structures and accessing in data-base system, IBM Systems Journal, 12,1 (1973), 30-93.

[SIBL74] Sibley, E.H., On the equivalence of data base systems, Proceedings of ACM SIGNOD 1974 Debate on Data Models, Ann Arbor, Michigan (May 1974), 43-76.

[SIBL77] Sibley, E.H. and Kershberg, L. Data architecture and data model considerations, Proceedings of AFIPS National Computer Conference, Dallas, Texas (June 1977), 85-96.

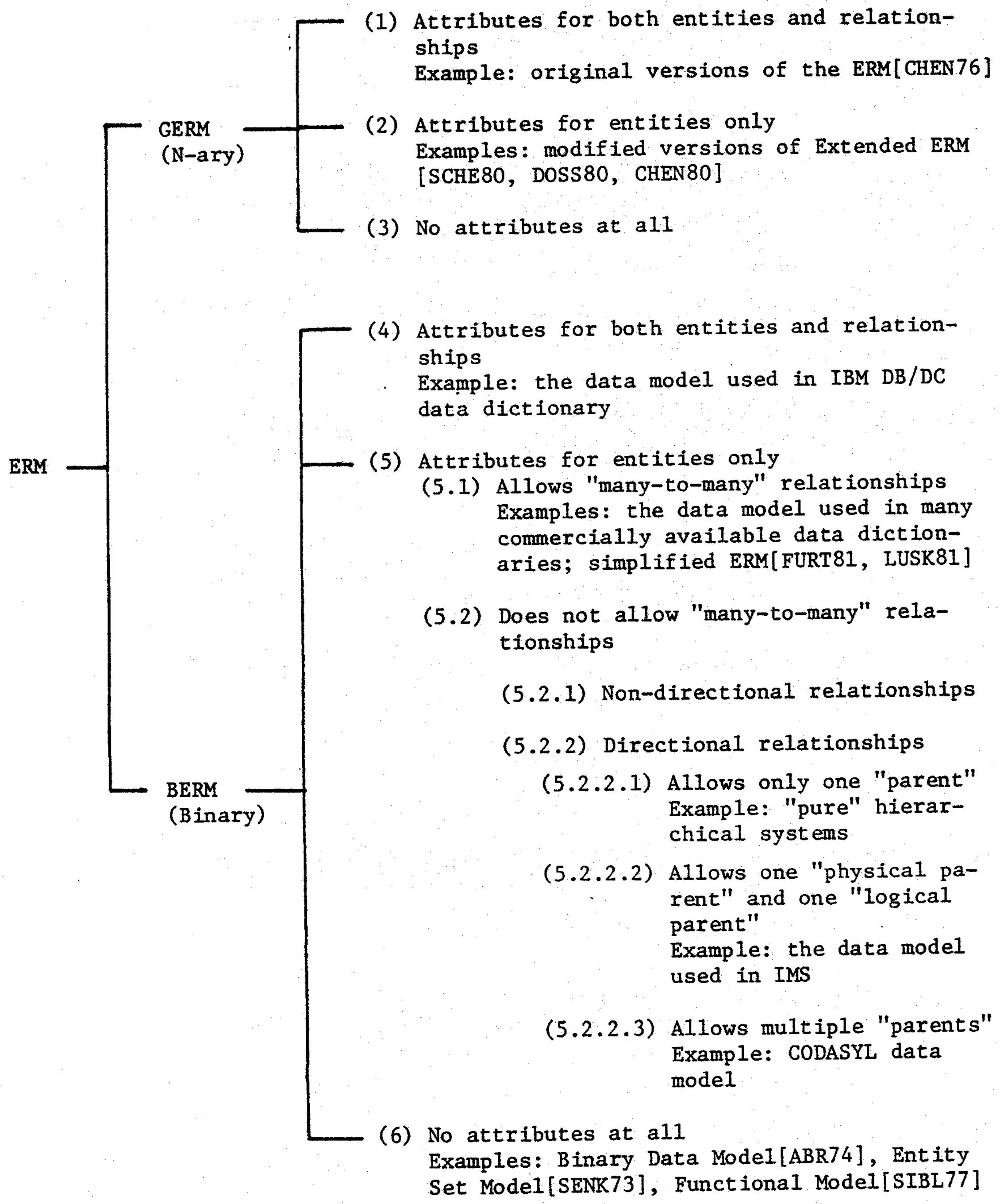

ERM
GERM (N-ary)
(1) Attributes for both entities and relationships
Example: original versions of the ERM[CHEN76]
(2) Attributes for entities only
Examples: modified versions of Extended ERM [SCHE80, DOSS80, CHEN80]
(3) No attributes at all
BERM (Binary)
(4) Attributes for both entities and relationships
Example: the data model used in IBM DB/DC data dictionary
(5) Attributes for entities only
(5.1) Allows "many-to-many" relationships
Examples: the data model used in many commercially available data dictionaries; simplified ERM[FURT81, LUSK81]
(5.2) Does not allow "many-to-many" relationships
(5.2.1) Non-directional relationships
(5.2.2) Directional relationships
(5.2.2.1) Allows only one "parent"
Example: "pure" hierarchical systems
(5.2.2.2) Allows one "physical parent" and one "logical parent"
Example: the data model used in IMS
(5.2.2.3) Allows multiple "parents"
Example: CODASYL data model
(6) No attributes at all
Examples: Binary Data Model[ABR74], Entity Set Model[SENK73], Functional Model[SIBL77]

Figure 1: A Framework for Entity-Relationship Models

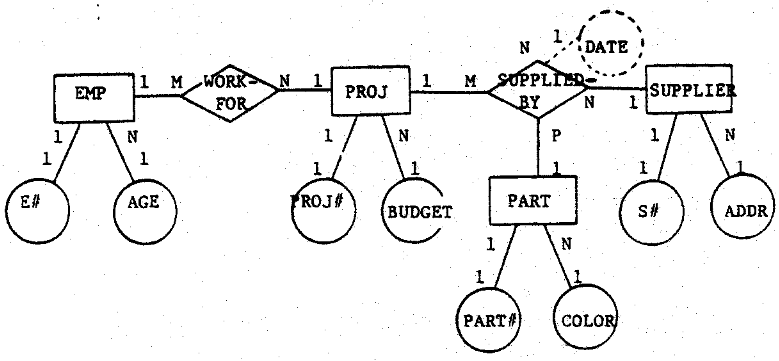


Figure 2: An ERD based on a GERM which allows attributes on both entities and relationships

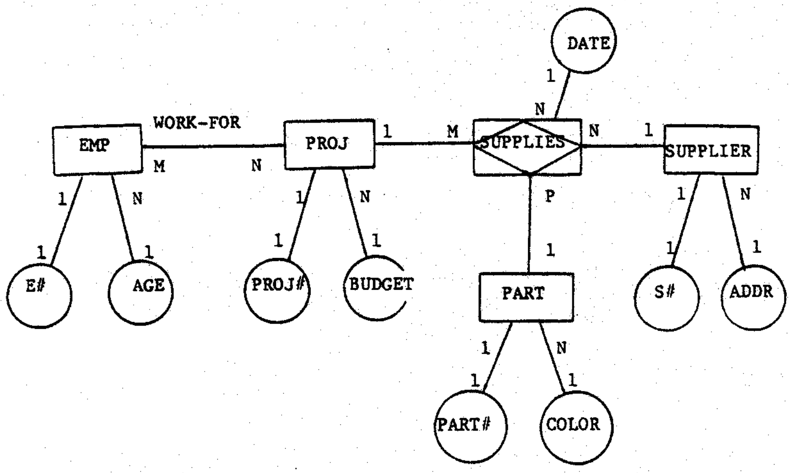


Figure 3(a): An ERD based on a GERM which allows attributes on entities only (Note that "DATE" is an attribute of a "high-level entity" of type "SUPPLIES".)

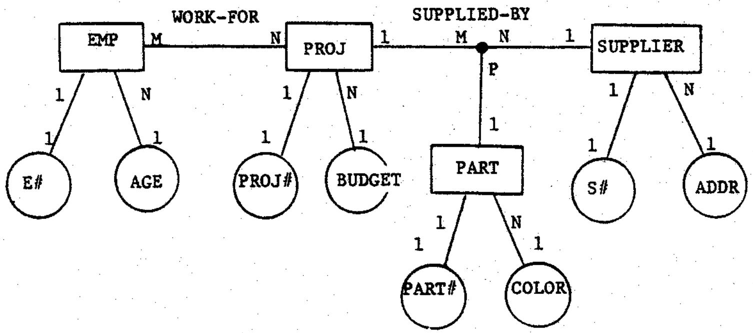


Figure 3(b): An ERD based on a GERM which allows attributes on entities only (Note that the "SUPPLIED" relationship does not have an attribute.)

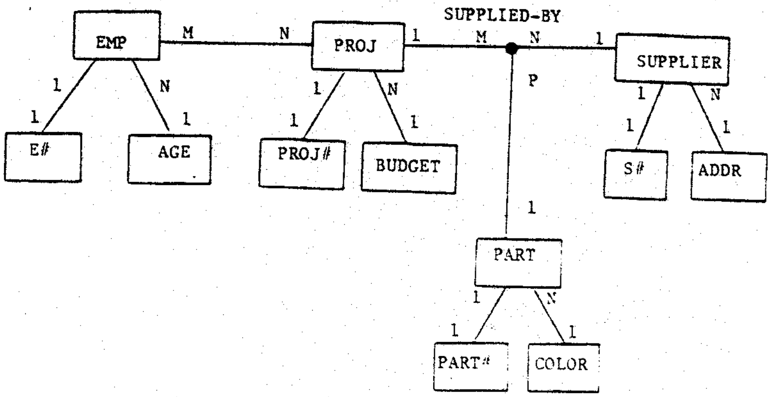


Figure 4: An ERD based a GERM with no attributes at all (This is a corresponding diagram of Figure 3(b).)

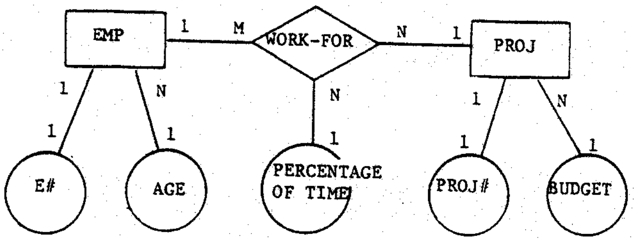


Figure 5: An ERD based on a BERM which allows attributes for both entities and relationships

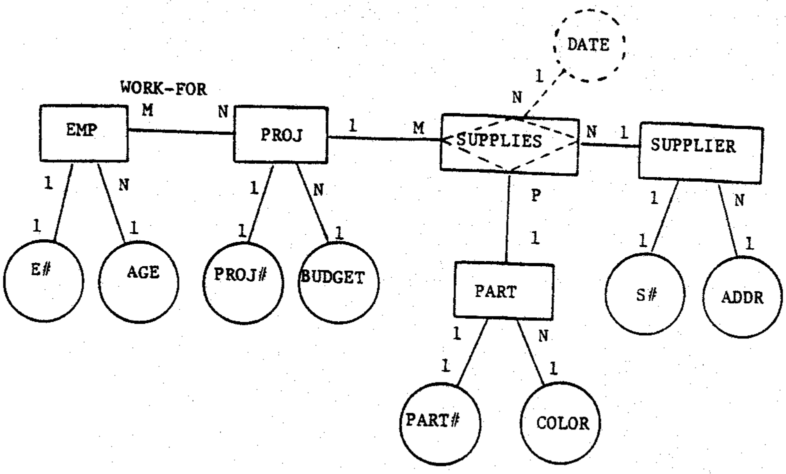


Figure 6: An ERD based on a BERM which allows attributes for entities only and allows "many-to-many" relationships

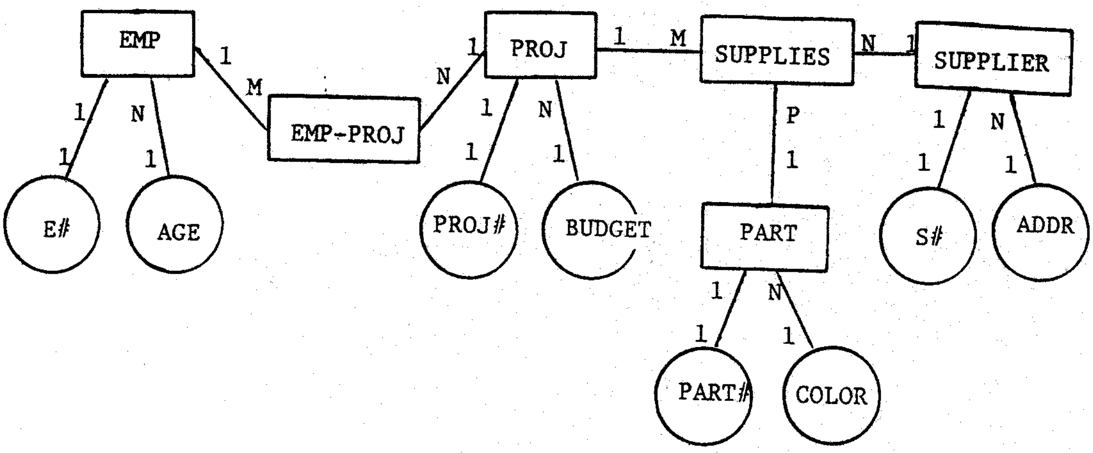
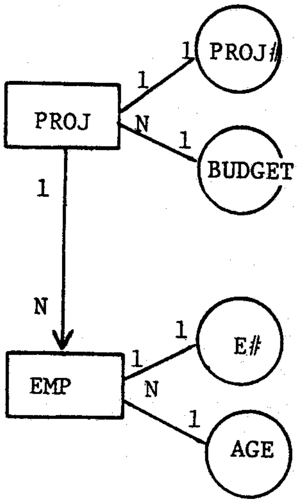


Figure 7: An ERD based on a BERM which allows only: (i) attributes for entities, (ii) "one-to-many" and "one-to-one" one-directional relationships



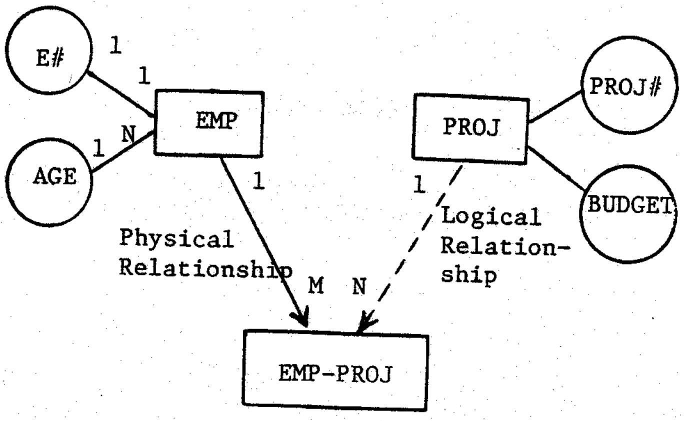


Figure 8: An ERD based on a BERM which allows only: (i) attributes for entities, (ii) "one-to-one" and "one-to-many" directional relationships, (iii) one "parent" for each child.

Figure 9: Same as stated in Figure 6 except that one "logical parent" and one "physical parent" are allowed

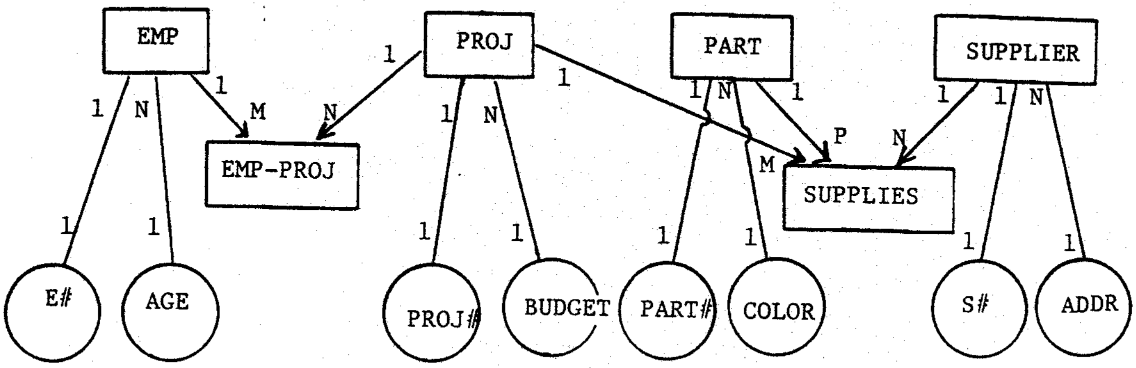


Figure 10: Same as stated in Figure 8 except that multiple "parents" are allowed

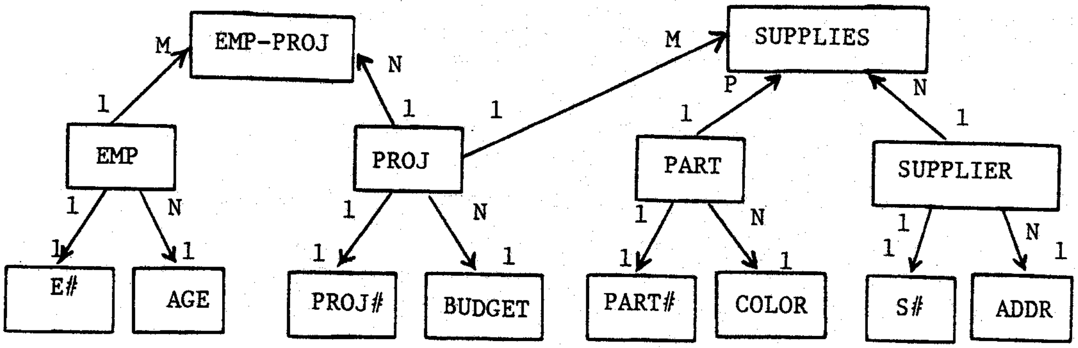


Figure 11: An ERD on a BERM in which no attribute is allowed

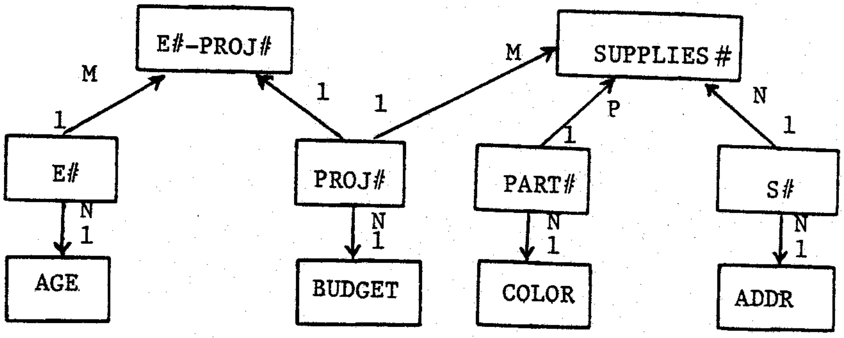


Figure 12: Another ERD on BERM in which no attribute allowed and the entity identifying attributes are used to identify entities themselves