Purdue University

Purdue e-Pubs

Department of Computer Science Technical Reports

Department of Computer Science

1977

Logical Data Base Design Principles for CODASYL Data Base Management Systems

Thomas I. M. Ho

Patrick A. Blosser

Report Number:

77-239

Ho, Thomas I. M. and Blosser, Patrick A., "Logical Data Base Design Principles for CODASYL Data Base Management Systems" (1977). *Department of Computer Science Technical Reports.* Paper 174.
https://docs.lib.purdue.edu/cstech/174

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.
Please contact epubs@purdue.edu for additional information.

LOGICAL DATA BASE DESIGN PRINCIPLES FOR CODASYL DATA BASE MANAGEMENT SYSTEMS

Thomas I. M. Ho
Purdue University
Computer Sciences Department
and
School of Management
West Lafayette, Indiana 47907

Patrick A. Blosser
Bell Laboratories
Holmdel, New Jersey 07733

CSD-TR 239

July 1977

LOGICAL DATA BASE DESIGN PRINCIPLES FOR CODASYL DATA BASE MANAGEMENT SYSTEMS

Thomas I. M. Ho

Purdue University
Computer Sciences Department
and
School of Management
West Lafayette, Indiana 47907

Patrick A. Blosser

Bell Laboratories
Holmdel, New Jersey 07733

Keywords: CODASYL, data base management system

Categories: 2.42, 3.50, 4.33

ABSTRACT

The role of the data base in an information system is characterized. The role of the data base warrants its use to represent relevant entities and relationships that exist in the organization supported by that data base. Unfortunately, no clear correspondence exists between the data definition facilities of available data base management systems and the organizational elements to be represented in the data base. Conventions for construction of a CODASYL-DBTG schema from a description of an organizational system are presented. The conventions are presented both in terms of a practical language for requirements statement and of a formal model for data definition.

INTRODUCTION

Systems analysis enables the translation of an unstructured organizational perspective of a problem into the rigorous hardware and software solutions available with computing technology. On one hand, the organization system is qualitative and unstructured. On the other hand, the computer system is technical and rigorous.

An information system is the interface between an organization system and a computer system [12]. The information system is a decoupling mechanism that provides a standard that enables organization system concepts to be expressed in a conceptual framework that is also compatible with computer system concepts.

An information system is composed of interacting subsystems:

1. Input subsystem
2. Output subsystem
3. Data base subsystem
4. Process subsystem.

The data base subsystem is of particular interest as a decoupling mechanism between the input and output subsystems. The input subsystem receives data from the environment. The output subsystem generates information to the environment. However, the output subsystem does not necessarily generate information at the same time nor at the same rate as the input subsystem receives data. Therefore, the data base subsystem is an inventory of data resources. Furthermore, the output subsystem does not necessarily generate information in a format that is identical with that of the data used to generate the desired information. Hence, the data base subsystem maintains a standard specification for data resources in order to decouple the incompatibilities between the input and output subsystems. The decoupling role of the data base subsystem in these respects motivates the residence of the data base subsystem in the storage subsystem of a computer system.

With respect to the organization system, the data base subsystem also functions as a decoupling mechanism. The various functional subsystems of an organization system are interacting subsystems that must communicate with one another to achieve the desired synergistic effect. Again, the data base subsystem serves as both an inventory and is a standard for the data resources that are generated by any functional subsystem and can be used by any other functional subsystem in pursuit of that subsystem's objectives. Similarly, the data base subsystem also decouples separate procedures and models within a single subsystem. However, it is the data base subsystem's role as a decoupling mechanism between functional subsystems that elevates it to its central role in an integrated information system.

In its role as a decoupling mechanism, the data base subsystem should therefore contain representations of the persons, objects, and events of interest to organizational activities. The elements of the data base subsystem that represent these persons, objects, and events are called **entities**. Furthermore, the data base subsystem should also contain representations of the relevant associations among the organizational persons, objects, and events. The elements of the data base subsystem that represent these associations are called **relationships** among the corresponding entities. The concepts of entity and relationship for data base definition have been proposed by both Kahn [16] and Chen [5].

In view of its central role in the information system, the data base subsystem is particularly important as an interface between the organization and computer systems. The data base subsystem must represent the persons, objects, and events in the organization system that determine the outcome of the actions and decisions performed by the information system. At the same time, the data base subsystem must reside in the storage subsystem of a computer system selected for implementation of the information system. The organization of the data base in the storage subsystem must conform to the hierarchy of data structures (file, record, group, element) available in computer systems. Unfortunately, there is no clear correspondence between the various organizational elements that are to be represented in the data base and the data structures that are available for implementing the data base. Even the hierarchy of data structures that are available with contemporary data base management systems (DBMS) do not clearly correspond to the various organizational elements. Hence, we are in need of a conceptual framework that establishes a set of guidelines and principles for the task of translating our qualitative and unstructured perception of organizational requirements into the technical and rigorous solutions that are available with computerized hardware and software. In particular, with respect to logical data base design, we need principles to enable a systems analyst to determine the contents of a data base from his study of the organizational system to be supported by that data base.

Logical data base design is the process that determines the composition of a data base and the logical relationships among the components that constitute the data base. The composition of a data base is characterized by grouping data elements into various record types. Logical relationships among the components of a data base are characterized by logical access paths among the occurrences of the various record types. The composition and logical relationships of a data base are commonly called a schema.

According to Bubenko et al. [4], logical data base design is performed on two levels. The **infological** level corresponds to the end-user level where information is referred to in problem-oriented and implementation-independent terms. The **datalogical** level corresponds to a level where one has decided on the data representation or schema. Therefore, the infological level corresponds to the information system and the datalogical level corresponds to the computer system. This dichotomy is also recognized by the ANSI/X3/SPARC Study Group on DBMS [1]. This group has designated the infological level as the **conceptual** level and the datalogical level as the **external** level.

Logical data base design infers a data base schema from the requirements of the organization to be served by that data base. In particular, these requirements express the data definition and manipulation of the desired applications. A Requirements Statement Language (RSL) has been advocated by Ho and Nunamaker [14] for the statement of information system requirements.

This paper presents principles for determining the contents of a data base that supports an organization. The desired result is a data base schema stated in the Data Definition Language (DDL) of a DBMS.

We recognize that the first step in data base design is determining **what** will be in the data base **before** determining **how** the contents of the data base will be logically or physically organized. Then, and only then, can other concerns be addressed. For example, the selection of an **optimal** data base design is constrained by the requirements of the organization to be supported by that data base. More attention has been given to the problem of optimizing datalogical structure. For example, Mitoma [17] and Hubbard and Raver [15] describe techniques for performance improvement of CODASYL-DBTG and IBM-IMS schemas, respectively. However, little concern has been displayed for the problem of inferring the initial feasible datalogical structure from which alternatives for subsequent optimization may be generated. Even Bubenko [4] emphasizes the generation of alternative schemas without indicating how to initially generate a canonical form.

Perhaps the most comprehensive treatments of infological-to-datalogical translation have been done by Gerritsen [9] and Chen [5]. Gerritsen advocates a functional approach to schema generation. The functional approach requires the infological level to be described in terms of the queries that will be posed to the data base. Such an approach risks incompleteness of the resulting schema due to the omission of queries from the infological level. The counterpart of the functional approach is the existential approach which requires the infological level to be described in terms of a model of the environment in which the data base exists. Chen advocates the existential description in terms of entities and relationships that exist in the environment. However, Chen's rules for infological-to-datalogical translation omit several concepts, e.g. identifiers, that must be considered in order to generate a complete schema. Probably, the ideal approach to schema generation is an existential-functional approach that requires an existential definition whose completeness for satisfying particular queries can be checked by computer-aided means in order to provide a complementary functional approach.

The rules for schema generation are expressed in terms of an information system model for interfacing the organization and computer systems. The model is characterized by both practical and formal models for data definition. The practical model is an RSL and the formal model is a mathematical model. Such models are consistent with the Conceptual Model of the ANSI/X3/SPARC Study Group on DBMS [1]. This approach enables consideration of data base design in the context of the organization to be supported by the data base. The relationship between the components of the data base and the elements of the application environment remains readily apparent. This approach is notable because it highlights the necessity of an initial canonical data definition that establishes the requirements that must be satisfied by the computerized data base. This approach is consistent with the structured programming tool of abstraction that first constructs a high-level abstract program that is a canonical form for lower-level programs that are inferred by a process of refinement. The initial step in this refinement process for determining the initial canonical data definition is described by Ho [13]. This paper describes the subsequent steps which are more fully described by Plosser [3].

THE PSL MODEL FOR REQUIREMENTS STATEMENT

The PSL under consideration in this study is the Problem Statement Language (PSL) developed by the ISDOS (Information System Design and Optimization System) Project. The major features of PSL are described by Teichroew and Hershey [19].

PSL enables the systems analyst to define the data base structure that describes the organization system to be supported by an information system. Data definition is accomplished by describing various objects and relationships among objects that model the environment in which the desired applications exist. PSL data object types include the ELEMENT, GROUP, ENTITY, and SET. PSL data relationships include specification of identifiers for an ENTITY, subsetting of a SET, and logical relationships between ENTITIES. Throughout this paper, PSL data objects and relationships will be capitalized.

THE PRISM MODEL FOR DATA DEFINITION

The abstract model for data definition is the PRISM (PRoperties of an Information System Model) model developed by Ho [11]. The major feature of PRISM data definition is the relational structure, a first normal form relation as defined by Codd [7]. The completeness of the relational model for data definition has been demonstrated by Codd [6]. A central concept of PRISM is the identifier set, a subset of the data names in a relational structure whose values enable the identification of occurrences of a relational structure.

Let R be the set of all number and character representations. Let U = {d<i>} be the data names in the information system being modeled by PRISM. Let a data_item be the ordered pair (d<i>,r), where d<i> is an element of U and r is an element of R, designating an occurrence_of_the_data_name d<i> with_value r.

A relational__structure D<h> is a set of data names. An occurrence_of_a_relational_structure D<h> is D = {(d<i>,r<i>): d<i> is an element of D<h> and r<i> is an element of R}. Then, v(d<i>,h,D) = r<i>. A data__base DB is a set of occurrences of relational structures. Then, DC<h> = {D is an element of DB: D is an occurrence of D<h>} is the set of occurrences of D<h> in DB.

The **identifier set** ID_h of a relational structure D_h is a subset of the data names in D_h with the following properties:

1. $ID_h(D) = \{(d_i, r_i)$ is an element of D: d_i is an element of $ID_h\}$ is the occurrence of ID_h for D.
2. $K_h = \{ID_h(D)$: D is an element of $DC_h\}$ is the set of all occurrences of ID_h in DB.
3. If $ID_{h'}$ is contained in ID_h, F is a mapping from DC_h to $DC_{h'}$ which to every element D_h in DC_h associates an element $D_{h'}$ in $DC_{h'}$ such that $ID_{h'}(D_{h'})$ is contained in $ID_h(D_h)$.
4. F is **one-one** if whenever D_i and D_j are elements of DC_h and $D_i \neq D_j$, then $F(D_i) \neq F(D_j)$.
5. F maps DC_h **onto** $DC_{h'}$ if for every $D_{h'}$ in $DC_{h'}$, there exists a D_h in DC_h such that $F(D_h) = D_{h'}$.
6. F maps DC_h one-one onto K_h.

The data names that belong to the identifier set are underlined in the definition of a relational structure.

THE CODASYL MODEL FOR DATA MANAGEMENT

The model for computerized data management is the network model developed by the CODASYL Data Base Task Group [6]. The major feature of the CODASYL model is a Set type, a logical relationship between an Owner Record type and one or more Member Record types. A Set type is depicted by a data structure diagram, a device developed by Bachman [2]. A data structure diagram consists of boxes representing the various record types and an arrow leading from the box representing the Owner record type to the Member record type(s). Each arrow is labelled with the name of the Set type on the right. To the left of the arrow, there appears the name of the sort key that can be used to search for occurrences of the Member record type(s). To create a data base reference point from which data base accesses originate, there exists a unique record type called the SYSTEM record type that contains no data elements. All other record types may consist of one or more data elements.

A Data Definition Language (DDL) may also be used to describe Set and Record composition. The DDL used in this study is described by Hershey [10]. A data structure diagram is illustrated in Figure 1 and the corresponding DDL representation is illustrated in Figure 2. Assume that the Record type OWNER-RECORD is composed of only one data element REC-ITEM and we shall disregard the composition of all other Record types.

The reader may wonder why the CODASYL model itself cannot be used as the model for statement of data definition requirements. Olle [18] describes several restrictions of the CODASYL model that require extra Record and Set types to represent certain types of logical relationships. Chen [5] states that such restrictions require the

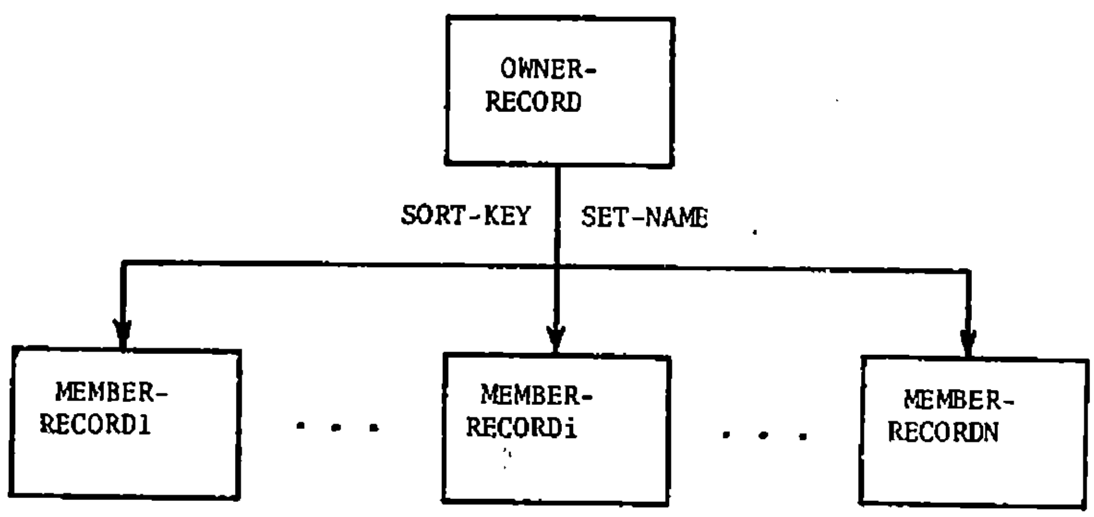
OWNER-
RECORD
SORT-KEY
SET-NAME
MEMBER-
RECORD1
. . .
MEMBER-
RECORDi
. . .
MEMBER-
RECORDN

Figure 1

definition of CODASYL structures in a manner that is not entirely consistent. We shall see that the PSL model for data definition corresponds closely to the organization system while enabling consistent usage of CODASYL concepts.

```
SET     SET-NAME      SORTED SORT-KEY
OWNER   OWNER-RECORD
MEMBER  MEMBER-RECORD1
           .
           .
           .
MEMBER  MEMBER-RECORDN

RECORD  OWNER-RECORD
ITEM    REC-ITEM
```

Figure 2

THE PSL MODEL FOR DATA DEFINITION

ELEMENT and GROUP

PSL facilities for data definition include the common elementary data structures: ELEMENT (ELE) and GROUP (GR). An ELEMENT is the basic unit of information and corresponds to a PRISM data name. A GROUP is a collection of ELEMENTS and/or other GROUPS. Codd [8] has shown that a GROUP can be represented by a set of relations that correspond to PRISM relational structures.

ENTITY

An ENTITY (ENT) CONSISTS OF ELEMENTS and/or GROUPS. Therefore, an ENTITY is a set of relational structures whose relational join represents the various ELEMENTS and/or GROUPS that are CONTAINED IN the ENTITY:

$$ENT\langle j\rangle = \{D\langle j,1\rangle, D\langle j,2\rangle, \ldots, D\langle j,m_j\rangle\}$$

where m_j is the number of relational structures needed to represent ENT<j>. An ENTITY is IDENTIFIED BY an ELEMENT or a GROUP. An ELEMENT or GROUP that IDENTIFIES ENT<j> is a PRISM identifier set: $\{ID\langle j,i\rangle: ID\langle j,i\rangle \subseteq ID\langle j,k\rangle$ for all k such that $1 \le k \le m_j\}$. Figure 3 is a PSL statement that describes the ENTITY for employees who have unique employee numbers. EMPL = {D<EMPL,1>, D<EMPL,2>, D<EMPL,3>} where

D<EMPL,1> = {EMPL-NO, EMPL-NAME, RACE, SEX}
D<EMPL,2> = {EMPL-NO, MONTH, DAY, YEAR}
D<EMPL,3> = {EMPL-NO, DEGREE, MAJOR}.

EMPL is IDENTIFIED BY ID<EMPL,1> = {EMPL-NO}.

```
ENTITY EMPL;
   CONSISTS EMPL-NO, EMPL-NAME, RACE, SEX, DATE-OF-BIRTH, EMPL-EDUC;
   IDENTIFIED BY EMPL-NO;
GROUP DATE-OF-BIRTH:
   CONSISTS MONTH, DAY, YEAR;
GROUP EMPL-EDUC:
   CONSISTS DEGREE, MAJOR;
ELEMENT  EMPL-NO,  EMPL-NAME,  RACE,  SEX,  MONTH,  DAY, YEAR, DEGREE,
MAJOR;
```

Figure 3

Figure 4 is a PSL statement that describes the ENTITY for employees whose employee numbers also indicate the department to which the employee belongs. Then, EMPL = {D<EMPL,1>, D<EMPL,2>} where

D<EMPL,1> = {EMPL-NO, DEPT-NO, EMPL-NAME}
D<EMPL,2> = {EMPL-NO, DEPT-NO, MONTH, DAY, YEAR}.

EMPL is IDENTIFIED BY ID<EMPL,1> = {EMPL-NO, DEPT-NO}. All components of a GROUP identifier are necessary to guarantee uniqueness among the occurrences of the ENTITY IDENTIFIED BY that GROUP.

```
ENTITY EMPL;
   CONSISTS EMPL-DEPT-ID, EMPL-NAME, DATE-OF-BIRTH;
   IDENTIFIED BY EMPL-DEPT-ID;
GROUP EMPL-DEPT-ID:
   CONSISTS EMPL-NO, DEPT-NO;
GROUP DATE-OF-BIRTH:
   CONSISTS MONTH, DAY, YEAR;
ELEMENT EMPL-NO, DEPT-NO, EMPL-NAME, MONTH, DAY, YEAR;
```

Figure 4

In the CODASYL model, each PSL ENTITY assumes the role of a record type. Each IDENTIFIED BY relationship is effected by the creation of one or more CODASYL Sets. If an ENTITY is IDENTIFIED BY an ELEMENT, a Set is created with the SYSTEM record type as the Owner, the ENTITY record type as the Member, and the identifier ELEMENT as the sort key. A prototype PSL statement for an ENTITY IDENTIFIED BY ELEMENTS appears in Figure 5a while the corresponding CODASYL DDL representation and data structure diagram appear in Figures 5b and 5c, respectively. If an ENTITY is IDENTIFIED BY a GROUP, a DUMMY record type is created for each ELEMENT CONTAINED IN the GROUP. The DUMMY record type is composed of one data element corresponding to the

ELEMENT for which the record was created. A hierarchy of Sets is created to partition the occurrences of the ENTITY into separate Set occurrences corresponding to the values of the elements in the GROUP that IDENTIFIES the ENTITY.

```
ENTITY ENT1;
   IDENTIFIED BY ELE1, ..., ELEN;
```

Figure 5a

```
SET     Si       SORTED ELEi   for 1 ≤ i ≤ N
OWNER   SYSTEM
MEMBER  ENT1
```

Figure 5b

Figure 5c

A prototype PSL statement for an ENTITY IDENTIFIED BY a GROUP appears in Figure 6a. The corresponding CODASYL DDL representation and data structure diagram appear in Figures 6b and 6c, respectively. Note that multiple occurrences of DUMMYi (2 ≤ i ≤ N) with identical values for ELEi will be created. To minimize the number of occurrences of DUMMY records, the hierarchy should be arranged with record type DUMMY1 corresponding to the element ELEi with the fewest number of occurrence values. Subsequent levels should be occupied by DUMMY record types in ascending order of number of occurrence values. Set S has no sort key because each occurrence of Set S has only one member occurrence. Set S can be eliminated with ENT1 then defined as the member of Set SN.

```
ENTITY ENT1;
   IDENTIFIED BY GR1;
GROUP GR1;
   CONSISTS ELE1, ..., ELEN;
```

Figure 6a

Let SYSTEM be equivalent to DUMMY0. For all i such that 1 ≤ i ≤ N

```
SET     Si       SORTED ELEi
OWNER   DUMMYi-1
MEMBER  DUMMYi

SET     S
OWNER   DUMMYN
MEMBER  ENT1

RECORD  DUMMYi
ITEM    ELEi
```

Figure 6b

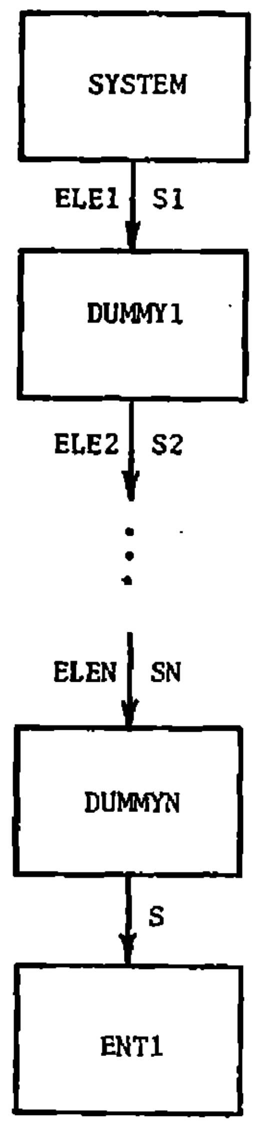

Figure 6c

SET and SUBSETTING CRITERIA

A SET CONSISTS OF one or more ENTITIES: $\{ENT_i: 1 \le i \le n\}$ where n is the number of ENTITIES in the SET. A SET is the collection of instances of the ENTITIES that are CONTAINED IN the SET: $\{OC\langle 1,1\rangle, \ldots, OC\langle 1,m_1\rangle, OC\langle 2,1\rangle, \ldots, OC\langle 2,m_2\rangle, \ldots, OC\langle n,1\rangle, \ldots, OC\langle n,m_n\rangle\}$. A SET may have SUBSETS which are characterized by SUBSETTING-CRITERIA (SSCA), ELEMENTS or GROUPS whose values distinguish the members of one SUBSET from other SUBSETS. Then, the data base may be searched for occurrences of ENTITIES that satisfy designated properties.

Figure 7 is a PSL statement that describes the SET of occurrences of the ENTITY EMPL. EMPL-SET = $\{OC\langle EMPL,1\rangle, OC\langle EMPL,2\rangle, OC\langle EMPL,3\rangle\}$ has SUBSETS whose members are characterized by their values for the relevant ELEMENT SUBSETTING-CRITERIA:

CHINESE = A $\cup$ B where

$$A = \{O\langle i\rangle \in OC\langle EMPL,1\rangle: v(RACE,\langle EMPL,1\rangle,O\langle i\rangle) = \text{"CHINESE"}\}$$

$$B = \{O\langle k\rangle \in OC\langle EMPL,k\rangle \text{ for } 1 < k \le 3: ID(O\langle i\rangle) \subseteq ID(O\langle k\rangle) \text{ where } O\langle i\rangle \in A\}.$$

MALE = A $\cup$ B where

$$A = \{O\langle i\rangle \in OC\langle EMPL,1\rangle: v(SEX,\langle EMPL,1\rangle, O\langle i\rangle) = \text{"MALE"}\}$$

$$B = \{O\langle k\rangle \in OC\langle EMPL,k\rangle \text{ for } 1 < k \le 3: ID(O\langle i\rangle) \subseteq ID(O\langle k\rangle) \text{ where } O\langle i\rangle \in A\}.$$

```
SET EMPL-SET;
   CONSISTS EMPL;
   SUBSETS CHINESE, JAPANESE, MALE, FEMALE;
SET CHINESE;
   DESCRIPTION;
      RACE = "CHINESE";
   CONSISTS EMPL;
   SSCA RACE;
   .
   .
   .
SET MALE;
   DESCRIPTION;
      SEX = "MALE";
   CONSISTS EMPL;
   SSCA SEX;
   .
   .
   .
```

Figure 7

Figure 8a is a PSL statement that describes the SET of occurrences of the ENTITY VEHICLE = {O<VEHICLE,1>, O<VEHICLE,2>} where

```
O<VEHICLE,1> = {SERIAL-NO, FUEL-ECON}
O<VEHICLE,2> = {SERIAL-NO, MAKE, NO-CYL}.
```

VEH-SET has SUBSETS whose members are characterized by their values for the relevant GROUP SUBSETTING-CRITERIA:

```
FORD-6-CYL = A U B where
A = {O<i> ε OC<VEHICLE,2>:   v(MAKE,<VEHICLE,2>,O<i>) = "FORD"
     and v(NO-CYL,<VEHICLE,2>,O<i>) = 6}
B = {O<k> ε OC<VEHICLE,1>:  IO(O<i>) = IO(O<k>) where O<i> ε A}.
```

All components of a GROUP SUBSETTING-CRITERIA are necessary to determine SUBSET membership of each occurrence of the ENTITY belonging to the SET that is being partitioned.

```
SET VEH-SET;
   CONSISTS VEHICLE;
   SUBSETS FORD-6-CYL, FORD-8-CYL, CHEV-6-CYL, CHEV-8-CYL;
SET FORD-6-CYL;
   DESCRIPTION;
      MAKE = "FORD" and NO-CYL = 6;
   CONSISTS VEHICLE;
   SSCA MAKE-ENGINE-CODE;
   .
   .
   .
ENTITY VEHICLE;
   CONSISTS SERIAL-NO, FUEL-ECON, MAKE-ENGINE-CODE;
   IDENTIFIED BY SERIAL-NO;
GROUP MAKE-ENGINE-CODE;
   CONSISTS MAKE, NO-CYL;
ELEMENT SERIAL-NO, MAKE, NO-CYL, FUEL-ECON;
```

Figure 8a

Finally, a PSL SET may not be homogeneous. A PSL SET CONSISTS OF more than one ENTITY type if the SET corresponds to a collection of ENTITY instances of similar, but not identical, characteristics. Figure 8b is a PSL statement that describes the SET of occurrences of the ENTITY types AUTO and TRUCK whose FUEL-ECON characteristic is of primary interest, but whose other characteristics are not identical.

```
SET VEH-SET;
   CONSISTS AUTO, TRUCK;
ENTITY AUTO;
   CONSISTS SERIAL-NO, FUEL-ECON;
ENTITY TRUCK;
   CONSISTS SERIAL-NO, FUEL-ECON, NO-AXLES;
ELEMENT SERIAL-NO, FUEL-ECON, NO-AXLES;
```

Figure 8b

In the CODASYL model, PSL SET representation depends on its homogeneity. If a SET SI is homogeneous, the SET is represented by a CODASYL Set SI as illustrated in Figures 5b and 5c.

If SET SI is non-homogeneous, SI is represented by two CODASYL Sets. A Set SI is created with the SYSTEM record type as the Owner and a record type NUBI as the Member. NUBI consists of one data element NUB-ITEMI. Then, a Set SSI is created with NUBI as the Owner and the record types ENT1, ..., ENTN as the Members. In each occurrence of Set SSI, all occurrences of ENTJ are owned by the occurrence of NUBI whose value for NUB-ITEMI equals j. Set SSI has no sort key. The members of SSI occur in first-in-first-out (FIFO) sequence. A prototype PSL statement for a SET appears in Figure 9a while the corresponding CODASYL DDL representation and data structure diagram appear in Figures 9b and 9c, respectively.

```
SET Si;
   CONSISTS ENT1, ..., ENTN;
```

Figure 9a

```
SET     Si      SORTED NUB-ITEMi
OWNER   SYSTEM
MEMBER  NUBi

SET     SSi      FIFO
OWNER   NUBi
MEMBER  ENT1
   .
   .
   .
MEMBER  ENTN

RECORD  NUBi
ITEM    NUB-ITEMi
```

Figure 9b
Non-homogeneous SET
without SUBSETTING-CRITERIA

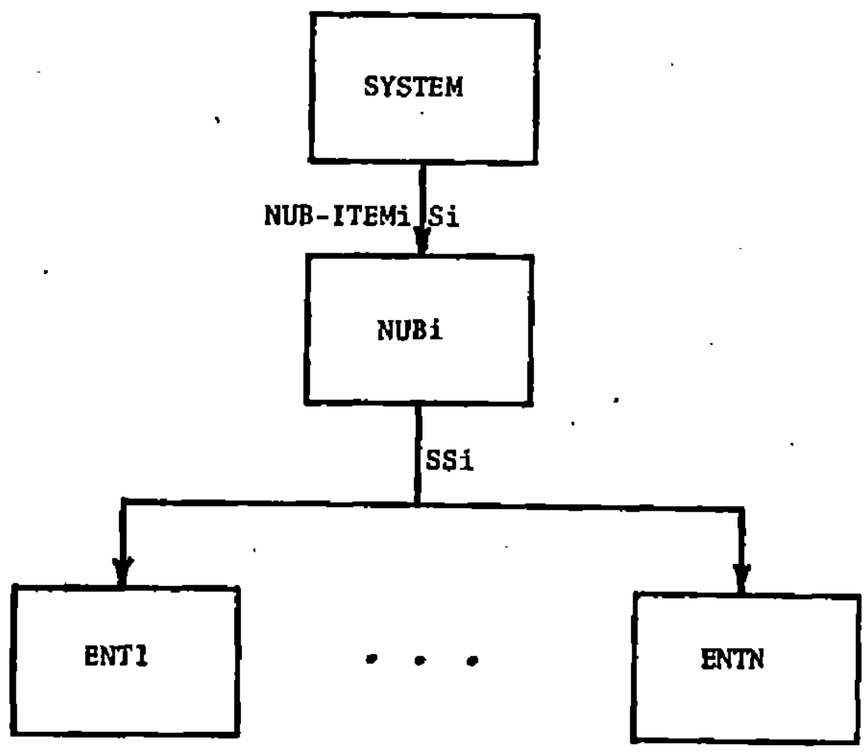

Figure 9c

Non-homogeneous SET
without SUBSETTING-CRITERIA

If a PSL SET has SUBSETTING-CRITERIA, its CODASYL representation also depends on the SET's homogeneity. If the SET is homogeneous, each GROUP SUBSETTING-CRITERION GRj is represented by a CODASYL Set SETi for each ELEMENT ELEi CONTAINED IN GRj. SETi has the record type DUMMYi-1 as the Owner and the record type DUMMYi as the Member, where the record type SYSTEM is equivalent to DUMMY0. Each record type DUMMYi consists of an ELEMENT ELEi CONTAINED IN GRj. Each SETi has ELEi designated as its sort key. Then, each occurrence of ENT is successively owned by those occurrences of DUMMYi that have a value for ELEi identical to the value of ELEi in the occurrence of ENT. Finally, Set SS has the record type DUMMYM as the Owner and the record type ENT as the Member. Of course, this solution requires the creation of occurrences of each DUMMYi with identical values of ELEi in order to accommodate occurrences of ENT that have identical values for one SUBSETTING-CRITERION and non-identical values for at least one other SUBSETTING-CRITERION. To minimize the number of occurrences of DUMMY records, the hierarchy should be arranged with record type DUMMY1 corresponding to the element ELEi with the fewest number of occurrence values. Subsequent levels should be occupied by DUMMY record types in ascending order of number of occurrence values. A prototype PSL statement for a SET with a GROUP SUBSETTING-CRITERION appears in Figure 10a while the corresponding CODASYL DDL representation and data structure diagram appear in Figures 10b and 10c, respectively. ELEMENT SUBSETTING-CRITERIA are represented in similar fashion since an ELEMENT is a GROUP that CONSISTS OF one ELEMENT. If the record type ENT does not participate in any other CODASYL Sets, note that the record type ENT need not contain items corresponding to the SUBSETTING-CRITERIA since the values of those items are implied by the Set occurrence membership of ENT.

```
SET S;
   SUBSETS S1, ...,SP;
   CONSISTS ENT1, ..., ENTN;
SET S1:
   CONSISTS ENT1, ..., ENTN;
   SSCA GR1;
   .
   .
   .
SET SP:
   CONSISTS ENT1, ..., ENTN;
   SSCA GR1;
GROUP GR1;
   CONSISTS ELE1, ..., ELEM:
```

Figure 10a

Let SYSTEM be equivalent to DUMMY0.

```
SET     SETi        SORTED ELEi   i = 1, ..., M
OWNER   DUMMYi-1
MEMBER  DUMMYi

SET     SS
OWNER   DUMMYM
MEMBER  ENT

RECORD  DUMMYi                    i = 1, ..., M
ITEM    ELEi
```

Figure 10b
Homogeneous SET
with SUBSETTING-CRITERIA

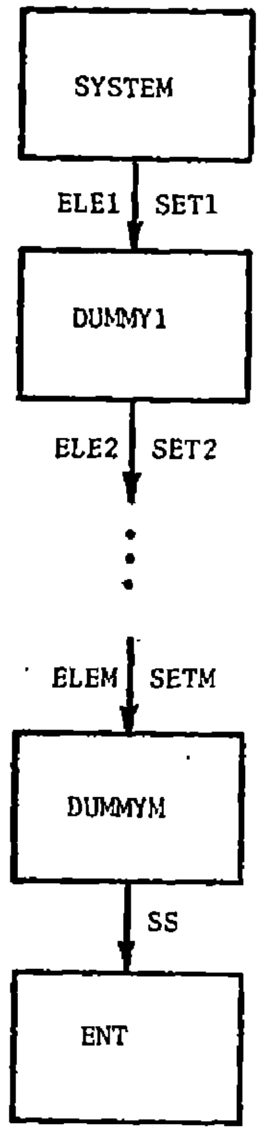

Figure 10c

Homogeneous SET
with SUBSETTING-CRITERIA

If SET S is non-homogeneous, the CODASYL representation is a combination of the two previous schemes. The corresponding CODASYL DDL representation and data structure diagram appear in Figures 11a and 11b, respectively.

Let NUB be equivalent to DUMMY0.

```
SET     S         SORTED NUB-ITEM
OWNER   SYSTEM
MEMBER  NUB

SET     SETi        SORTED ELEi i = 1, ..., M
OWNER   DUMMYi-1
MEMBER  DUMMYi

SET     SS
OWNER   DUMMYM
MEMBER  ENT1
    .
    .
    .
MEMBER  ENTN

RECORD  NUB
ITEM    NUB-ITEM
```

Figure 11a
Non-homogeneous SET
with SUBSETTING-CRITERIA

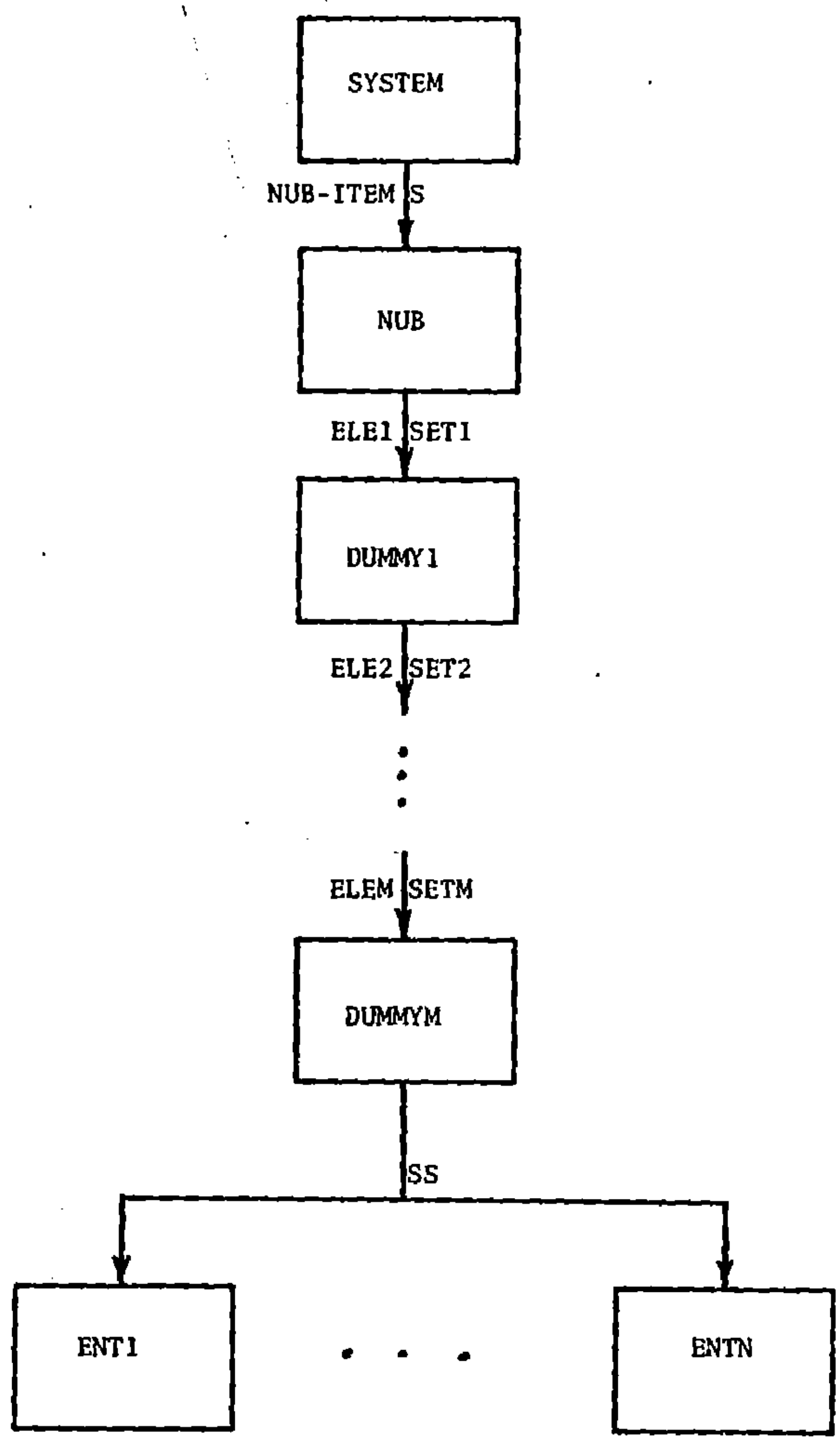

Figure 11b

Non-homogeneous SET
with SUBSETTING-CRITERIA

RELATION

A RELATION is a relationship between one ENTITY and another ENTITY. Consider the following RELATION:

```
RELATION R;
   BETWEEN ENTj AND ENTk;
   CONNECTIVITY M TO N;
   ASSOCIATED-DATA ELE1, ..., ELEm, GR1, ..., GRN;
```

For each occurrence of ENTj, the RELATION R specifies a subset of the occurrences of ENTk that is characterized by the occurrence of ENTj. The concept of a RELATION is especially useful for avoiding data redundancy and for identifying all occurrences of an ENTITY that have some property in common. In terms of the PRISM model, R = {ID<j>/FROM ID<k>/TO} where ID<i> IDENTIFIES ENTi. Note that R' = {ID<k>/FROM ID<j>/TO} is a DIFFERENT RELATION.

The CONNECTIVITY M TO N of RELATION R indicates that N is the maximum size of each subset of occurrences of ENTk that is characterized by each occurrence of ENTj while M is the maximum number of occurrences of ENTj that characterizes each occurrence of ENTk. In terms of the PRISM model, CONNECTIVITY M TO N means that

> For every O<k> ∈ OC<k>, there exists at_most M occurrences of O<j> ∈ OC<j> such that {ID(O<j>)/FROM, ID(O<k>)/TO} ∈ OC<R>.
>
> and
>
> For every O<j> ∈ OC<j>, there exists at_most N occurrences of O<k> ∈ OC<k> such that {ID(O<j>)/FROM, ID(O<k>)/TO} ∈ OC<R>.

ASSOCIATED-DATA are ELEMENTS or GROUPS that describe attributes of a RELATION. If ELEi is ASSOCIATED-DATA of RELATION R, then

R = {ID<j>/FROM, ID<k>/TO, ELEi}.

If GRi = {D<i>} is ASSOCIATED-DATA of RELATION R, then

R = {ID<j>/FROM, ID<k>/TO, {D<i>}}.

Figure 12 is a PSL statement that describes the RELATION SUPPLIES between the ENTITY VENDOR and the ENTITY PART. SUPPLIES = {VENDOR-NO/FROM, PART-NO/TO, PART-PRICE} identifies all occurrences of PART that have the common property of being supplied by the same VENDOR. In this case, a PART may be supplied by many VENDORS and a VENDOR may supply many parts. Finally, PART-PRICE describes the price charged by a particular VENDOR for a particular PART.

```
ENTITY VENDOR;
   CONSISTS VENDOR-NO, VENDOR-NAME;
   IDENTIFIED BY VENDOR-NO;
ENTITY PART;
   CONSISTS PART-NO, PART-NAME;
   IDENTIFIED BY PART-NO;
RELATION SUPPLIES;
   BETWEEN VENDOR AND PART;
   CONNECTIVITY MANY TO MANY;
   ASSOCIATED-DATA PART-PRICE;
```

Figure 12

In the CODASYL model, a RELATION's representation is dependent on its characteristics. Consider the prototype PSL statement for a RELATION in Figure 13.

```
RELATION R;
   BETWEEN ENT1 AND ENT2;
   CONNECTIVITY M TO N;
   ASSOCIATED-DATA ASSOC-DATA;
```

Figure 13

If ENT1 is identical to ENT2, R is represented by two CODASYL Sets. Set R1 is defined with record type ENT1 as the Owner and record type NUBR as the Member. Also, Set R2 is defined with NUBR as the Owner and ENT1 as the Member. If the RELATION has any ASSOCIATED-DATA, the associated ELEMENTS and/or GROUPS are contained in NUBR. The corresponding CODASYL DDL representation and data structure diagram appear in Figures 14a and 14b, respectively.

```
SET      R1
OWNER    ENT1
MEMBER   NUBR

SET      R2
OWNER    NUBR
MEMBER   ENT1

RECORD   NUBR
ITEM     ASSOC-DATA
```

Figure 14a
ENT1 identical to ENT2

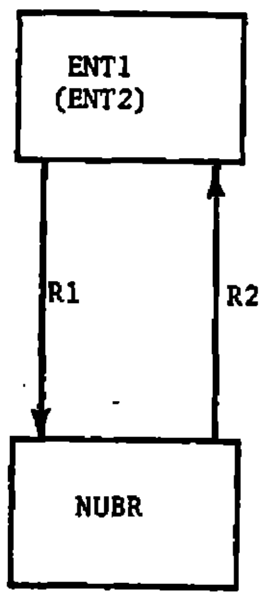

Figure 14b
ENT1 identical to ENT2

If ENT1 and ENT2 are not identical, the CONNECTIVITY of R determines its CODASYL representation. If M > 1 and N > 1, R is represented by two CODASYL Sets. Set R1 is defined with record type ENT1 as the Owner and record type NUBR as the Member. Also, Set R2 is defined with ENT2 as the Owner and NUBR as the Member. If the RELATION has any ASSOCIATED-DATA, the associated ELEMENTS and/or GROUPS are contained in NUBR. The corresponding CODASYL DDL representation and data structure diagram appear in Figures 15a and 15b, respectively.

```
SET     R1
OWNER   ENT1
MEMBER  NUBR

SET     R2
OWNER   ENT2
MEMBER  NUBR

RECORD  NUBR
ITEM    ASSOC-DATA
```

Figure 15a
ENT1 is not identical to ENT2
M > 1 and N > 1

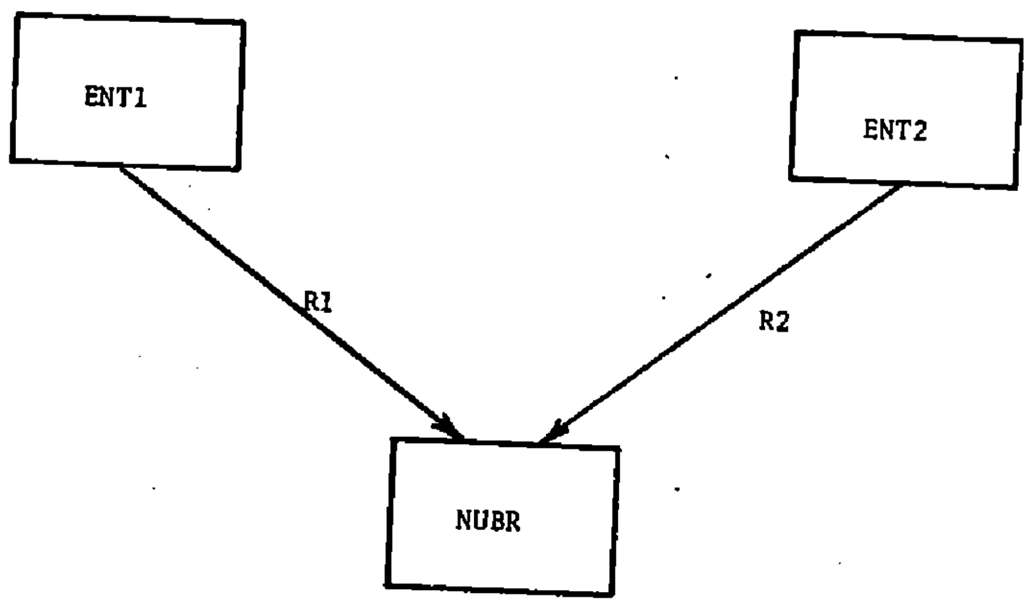

Figure 15b

M > 1 and N > 1

ENT1 is <u>not</u> identical to ENT2

If the complementary RELATION R' has also been defined:

```
RELATION R';
   BETWEEN ENT2 AND ENT1;
   CONNECTIVITY N TO M;
   ASSOCIATED-DATA ASSOC-DATA;
```

it is **not** necessary to define its CODASYL representation since the resulting data structure would be logically equivalent with the data structure resulting from RELATION R.

If M equals 1 and ENT1 is **not** identical to ENT2, the CODASYL representation is simple as long as R has **no** ASSOCIATED-DATA. A single CODASYL Set R is defined with record type ENT1 as the Owner and record type ENT2 as the Member. The corresponding CODASYL DDL representation and data structure diagram appear in Figures 16a and 16b, respectively.

```
SET     R
OWNER   ENT1
MEMBER  ENT2
```

Figure 16a
ENT1 is **not** identical to ENT2 and M = 1

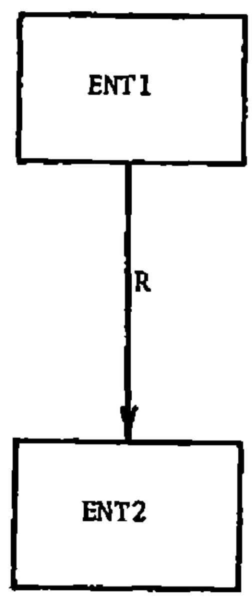

Figure 16b

ENT1 is _not_ identical to ENT2

and M = 1

The complementary RELATION R' is represented in the same way as RELATION R. The CODASYL Data Manipulation Language (DML) infers the current Owner occurrence whenever any Member occurrence is designated as the current Member of a CODASYL Set. In this way, the CODASYL DML provides the capability to determine the occurrence of ENT1 to which any occurrence of ENT2 is related by RELATION R'.

If M equals 1 and ENT1 is **not** identical to ENT2, the existence of ASSOCIATED-DATA introduces another consideration into CODASYL representation. Suppose there exists another RELATION R" BETWEEN some ENTi and ENT2 where i ≠ 1. For example, consider the following PSL statement:

```
RELATION REQUESTED-BY;
   BETWEEN PRODUCT AND ORDER;
   CONNECTIVITY 1 TO MANY;
   ASSOCIATED-DATA QUANTITY-REQUESTED;
RELATION PLACES;
   BETWEEN CUSTOMER AND ORDER;
   CONNECTIVITY 1 TO MANY;
```

According to the previously described procedure, this statement could be represented by two CODASYL Sets:

```
SET     REQUESTED-BY
OWNER   PRODUCT
MEMBER  ORDER

SET     PLACES
OWNER   CUSTOMER
MEMBER  ORDER
```

The ASSOCIATED-DATA QUANTITY-REQUESTED would then be contained in the record ORDER. However, it is likely that the Set PLACES may be used to find all the ORDERs of a CUSTOMER without needing to know about the QUANTITY-REQUESTED by each ORDER. In other words, the QUANTITY-REQUESTED need **only** be known in the context of the PRODUCT that was requested by an ORDER. In this case, it would be advisable to redefine the Set REQUESTED-BY to separate the ASSOCIATED-DATA from the ORDER record so that the QUANTITY-REQUESTED would be accessed **only** when the PRODUCT REQUESTED-BY an ORDER was of interest.

In summary, suppose R has ASSOCIATED-DATA with M equal to 1 and ENT1 **not** identical to ENT2. If there does **not** exist another RELATION R" BETWEEN some ENTi and ENT2 where i ≠ 1, R is represented as Figure 16a with ASSOC-DATA contained in record ENT2. If R" exists, R is represented by two CODASYL Sets. Set R1 is defined with record type ENT1 as the Owner and record type NUBR as the Member. Also, Set R2 is defined with NUBR as the Owner and ENT2 as the Member. If the RELATION has any ASSOCIATED-DATA, the associated ELEMENTS and/or GROUPS are contained in NUBR. The corresponding CODASYL DDL representation and data structure diagram appear in Figures 17a and 17b, respectively.

```
SET     R1
OWNER   ENT1
MEMBER  NUBR

SET     R2
OWNER   NUBR
MEMBER  ENT2

RECORD  NUBR
ITEM    ASSOC-DATA
```

Figure 17a
R has ASSOCIATED-DATA and R" exists
M = 1 and ENT1 is **not** identical to ENT2

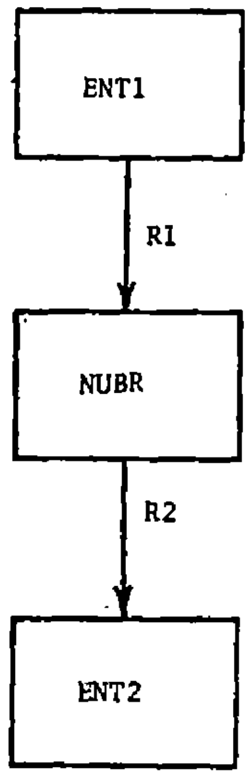

Figure 17b

R has ASSOCIATED-DATA and R" exists
M = 1 and ENT1 is not identical to ENT2

CONCLUSION

We have described a much-needed perspective of the data base subsystem of an information system. This perspective enables definition of the data base in terms of the characteristics of the organization system to be related to design of the data base in terms of the data management software available in the computer system.

REFERENCES

1. ANSI/X3/SPARC Study Group on DBMS. Interim Report. Doc. No. 7514TS01, CBEMA, Washington, DC (February 1975).

2. Bachman, C. W. Data structure diagrams. Data Base 1, 2 (Summer 1969), pp. 4-10.

3. Blosser, P. A. An automatic system for application software generation and portability. Ph.D. Diss., Computer Sciences Dept., Purdue U., West Lafayette, IN (May 1976).

4. Bubenko, J. A. Jr.; Berild, S.; Lindencrona-Ohlin, E.; and Nachmens, S. From information requirements to DBTG data structures. Proc. Conf. on Data: Abstraction, Definition and Structure 1976, ACM, New York, pp. 73-85.

5. Chen, P. P. S. The entity-relationship model: toward a unified view of data. Trans. Database Systems 1,1 (March 1976), pp. 9-36.

6. CODASYL. Data Base Task Group Report. ACM, New York (April 1971).

7. Codd, E. F. A relational model of data for large shared data banks. Comm. ACM 13, 6 (June 1970), pp. 377-387.

8. Codd, E. F. Normalized data base structure: a tutorial. Proc. 1971 ACM SIGFIDET Workshop: Data Description, Access and Control, ACM, New York, pp. 1-17.

9. Gerritsen, R. A preliminary system for the design of DBTG data structures. Comm. ACM 18, 10 (October 1975), pp. 551-557.

10. Hershey, E. A. III. A data base management system for PSA based on DBTG 71. ISDOS Working Paper No. 88, Dept. Industrial and Operations Eng., U. Michigan, Ann Arbor (September 1973).

11. Ho, T. I. M. Toward a formal theory for the requirements statement, analysis, and design of information systems. Ph.D. Diss., Computer Sciences Dept., Purdue U., West Lafayette, IN (December 1974).

12. Ho, T. I. M. Systems analysis perspectives. Proc. 14th Annual Computer Personnel Research Conference 1976, ACM, New York, pp. 12-19.

13. Ho, T. I. M. Data base concepts for systems analysis. CSDTR No. 219, Computer Sciences Dept., Purdue U., West Lafayette, IN (January 1977).

14. Ho, T. I. M. and Nunamaker, J. F. Jr. Requirements statement language principles for automatic programming. Proc. 1974 ACM National Conference, ACM, New York, pp. 279-288.

15. Hubbard, G. and Raver, N. Automating logical file design. Proc. First Annual Conference on Very Large Data Bases 1975, ACM, New York, pp. 227-253.

16. Kahn, B. K. A method for describing information required by the database design process. Proc. 1976 ACM-SIGMOD Intl. Conf. on Mgmt. of Data, ACM, New York, pp. 53-64.

17. Mitoma, M. F. and Irani, K. B. Automatic data base schema design and optimization. Proc. First Annual Conference on Very Large Data Bases 1975, ACM, New York, pp. 286-321.

18. Olle, T. W. Current and future trends in data base management systems. Proc. Information Processing 74, North Holland Publishing, Amsterdam, pp. 998-1006.

19. Teichroew, D. and Hershey, E. A. PSL/PSA: a computer-aided technique for structured documentation and analysis of information processing systems. IEEE Trans. Software Eng. SE-3, 1 (January 1977), pp. 41-48.