International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 6, Issue 4 (March 2013), PP. 51-60

E-R Model to an Abstract Mathematical Model for Database Schema using Reference Graph

Subhrajyoti Bordoloi[1], Bichitra Kalita[2]
Department Of Computer Applications, Assam Engg. College, Guwahati-13, Assam.

Abstract:- Graphs have many applications in the field of computer science. As a data structure it plays a very important role in database management system design. In this paper we discuss how the references among the database tables can be easily described by graphs. In the new approach, unlike E-R model, the reference graph model can describe the references occurred among the database tables or among various databases of diverse platforms more precisely and clearly. The reference graphs, unlike the schema diagrams, do not show the details about the attributes of the entities, rather it gives the references available among the entities and the attributes through which these references are made. Reference graphs model, derived from the E-R model or from the intermediate graph, directly developed by analyzing the context system, is used to develop an abstract mathematical model of the database schema, from which we develop the Database Requirements Specifications (DRS). The DRS can be used as a design document in RDBMS and OODBMS paradigm.

Key Words:- E-R Diagrams, EER, SRS, DRS, Reference Graph (E-R graph), FD.

I. INTRODUCTION

It is found that Entity-Relationship diagram plays a vital role in structured analysis and conceptual modelling. Entity-relationship (ER) model is a commonly used abstract representation of database structure. In building a relational database the first step is the creation of an ER model. An ER model is a representation of a database structure. It is not about the contents of the database. There are many classification of E-R model depending of their definition and representation [1], and a comparison of various models is discussed in by Il-Y. Song, et al.[2]. They have discussed about how E-R models are developed to define database. These are basically graph based data model. The entities in a system and their relationships are modelled by using graph based models. Graph based approaches are widely used in database management, data retrieval and data mining. The E-R models represent the entities and their relationship only. P. S. Dhabe, et al. [3] has proposed a new model called 'Articulated Entity Relationship' (AER) as an extension of Entity Relationship (ER) diagram to accommodate the Functional Dependency (FD) information. Updating the existing attributes may lead to inconsistent description of FD information and attribute description of entities. [3]. D. Heckerman, et al. [4] proposed a model that incorporated the probabilistic relationship into E-R models. A. Sarkar [5] in his paper proposed a graph based approach for modelling of irregular, heterogeneous, hierarchical and non-hierarchical semi-structured data at the conceptual level in object oriented paradigm.

Many graph based approaches are being used in knowledge and data engineering. Y.Yan***g*** [6], in his paper, proposed a new recommendation technique based on reference graph which can recommend research paper in an effective manner. The reference graph based recommendation technique is more effective [6]. M. Graves et al. [8] proposed a graph representation of Genome database. M. Tavana, et al. [9] has proposed an algorithm for re-clustering of E-R models. They use a method for clustering of entities to form small interrelated modules of the E-R diagram consisting of very large number of entities and relationships. R. Angels at el [10], in their paper presented a survey to present the work that has been conducted in the area of graph database modelling .There are many automated tools for designing database schema.[11][12]. L. Simasatitkul et al, [11] in their paper, proposed a tool for generating database schema from EER diagram, written in XML format. The goals of this tool are transforming an EER diagram to relational database schema and to specify relational database constraints in the relations.

In this paper, we are presenting a simple method for designing database schema which will help the students, designers and software developers.

II. OBJECTIVE

Representing the E-R model as a graph is very helpful in database design and maintenance. If the structure of a table is to be changed to meet the future requirements (adding new attributes/ removing existing attributes), the number of out-degree and the number of in-degree of a node representing a table will give the number and names of the tables that may be affected due to the changes to be made. In this paper we are proposing a comprehensive method for analysis and design of database schema, which can be used as a design document in RDBMS and OODBMS paradigm.

III. REFERENCE GRAPH

The E-R model can be represented as a graph G= {V, E}, called reference graph, where V is the set of entities and E is the set of relationships. The relationships are represented as directed graphs. . *The set V also includes the m: n relationships.* There is a set, K_{SYS}, the key set, which contains the primary keys of all sets in the system. A set representing an entity or a relationship in the set V contains two sub sets - the set of identifying attributes (keys), K and the set of describing attributes, D.

A. Unary Relationship

In unary relationship same one entity is associated with itself. It is very important to specify the role of the participants in the relationship. The reference graph for unary relationship is shown in Fig.1

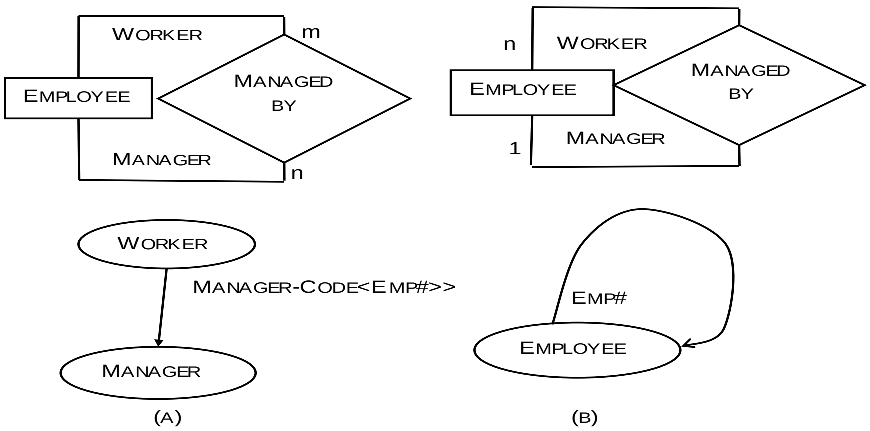


Figure 1: Unary relationships and their reference graphs

B. Binary Relationship

The n: 1 relationship is represented by an edge from n-side to the 1-side. For example, we just draw an edge from STUDENT to BRANCH as shown in Fig-2(b).For a 1:1 relationship we can draw an edge from one entity to the other as shown in the Fig-2(a). For each **m:n** relationship a new vertex is created. For example, we create a new vertex SCORE for the <TAKE> relationship in the reference graph as shown in Fig.2 (c).

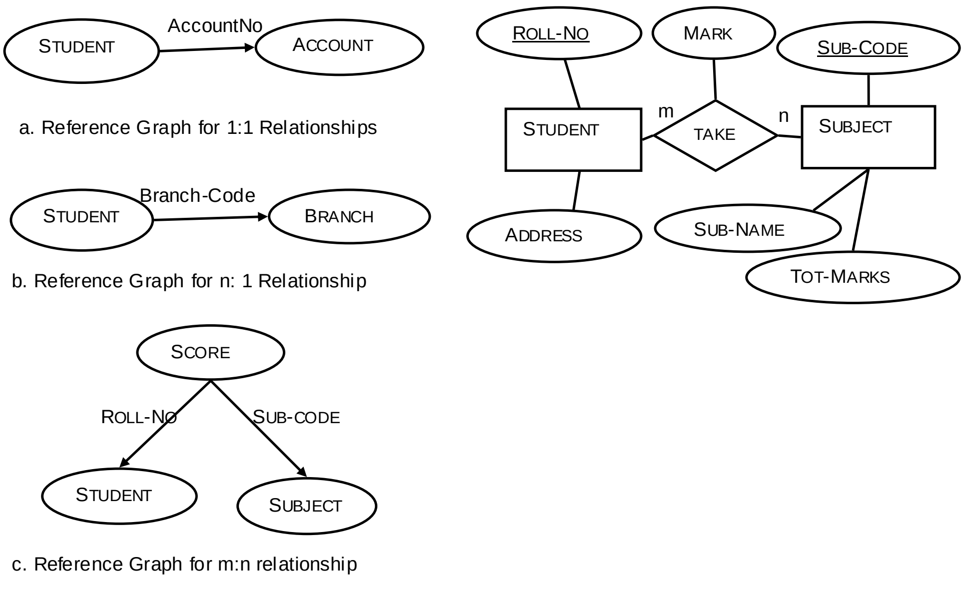
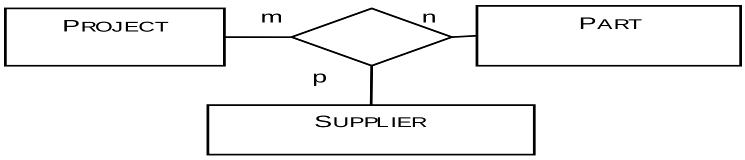


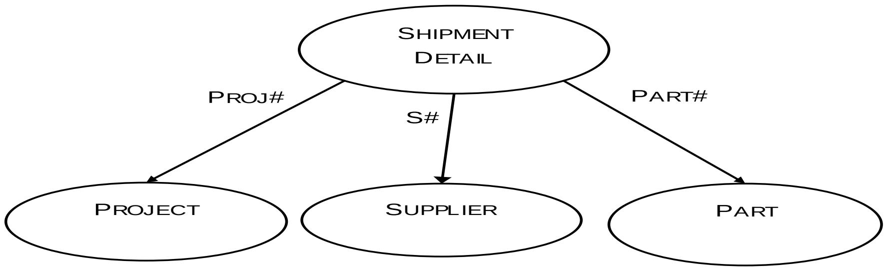
Figure 2: Binary relationships and their reference graphs

C. Ternary Relationship

A ternary relationship and its corresponding reference graph are shown in Fig.3 (a) and Fig.3 (b) respectively.



(a) Ternary Relationship



(b) Rreference Graph

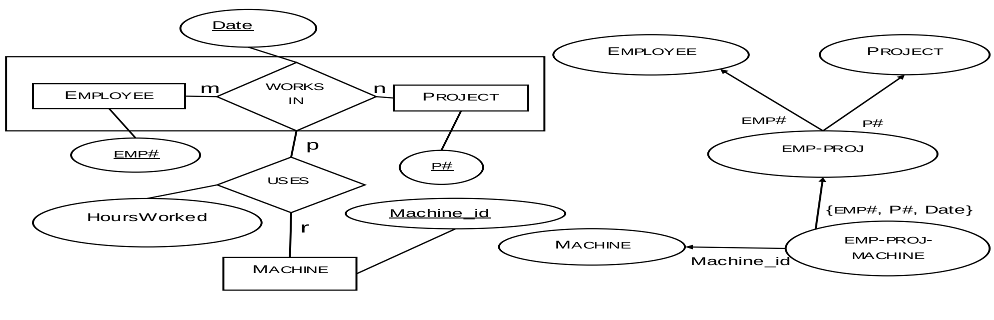
Figure 3: Ternary relationship and its reference graph

D. Specializations and Aggregations

For specialization, in the reference graph we create a separate set for each specialized entity assuming that there is a reference from the specialized entity to the generalized entity as shown in Fig.4 (a). For mapping of specialization types the concepts discussed in by Lisa Simasatitkul et al, *[11]* are used. A relation is created for each subclass and all the simple attributes of subclass are also created. The primary key of subclass is derived from primary key of super class.

. For aggregation, we create a set for the relationship between the sets in the aggregation and this new set can be considered as a regular set in the system. An example is shown in Fig.4 (b).

(a) Generalization and its Reference Graph



(b) Aggregation and its Reference Graph

Figure 4: Specialization and Aggregation

In an E-R graph, also called as *Reference Graph*, the vertex is nothing but a database table and the edges are nothing but the associations among the tables. In this context we propose the following theorems.

Theorem 1: *In a reference graph, the out-degree of a vertex represents the number of tables (vertices) it has referenced .The edge names represent the fields or attributes through which it has referred to other tables (vertices).*

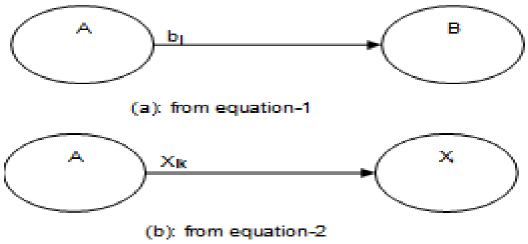


Figure 5: Graphs for Equations 1 and 2

Proof: [The vertices of the reference graph are nothing but sets of attribute types. For each reference graph there is a key, set K]

Let A and B are two sets and

A= {b_i, a_1, a_2, a_3, ...,a_n }

B= {b_1, b_1, b_2, b_3, ...,b_m }

Now, A∩B= {bi} 1≤ I ≤ m ----- (1).

Equation (1) means that there is a reference from A to B through bi, (bi Є B) & (b_i Є K), and it is graphically represented in the reference graph as shown in fig.-5(a). Now, let us assume that there exists set A and a set of set X={X_1, X_2, X_3 .... , X_k } in the system. By equation (1) we get that, if A∩ X_1={ x_{1j}}, 1≤ j ≤m , then there must be an outgoing edge from A to the set X1 in the reference graph of the system. If there is at least one reference from A to any set in X, then the following inequality must hold.

$$\cup_{i=1}^{k}(A \cap X_i) \neq \ldots\ldots\ldots(2)$$

By equations (1) and (2) we get that

if $\left|\cup_{i=1}^{k}(A \cap X_i)\right| = v$,v>0 then there are v number of references from A to set X which is represented by v number of outgoing edges from A in the reference graph.

Example: In Fig. 2(b) the vertex SCORE has referenced to the vertex STUDENT by the attribute 'rollno' and to the vertex SUBJECT by 'sub-code'. This means that the table has reference to two tables- STUDENT and SUBJECT.

Theorem 2: *In a reference graph, the in-degree of a vertex represents the number of tables (vertices) where the attributes, represented by the edges, of the table (vertex) are foreign keys. [Proof is similar to Theorem 1.]*

Example: In Fig.-2(a) the attribute BR-CODE of table (vertex) BRANCH is a foreign key in the table (vertex) STUDENT.

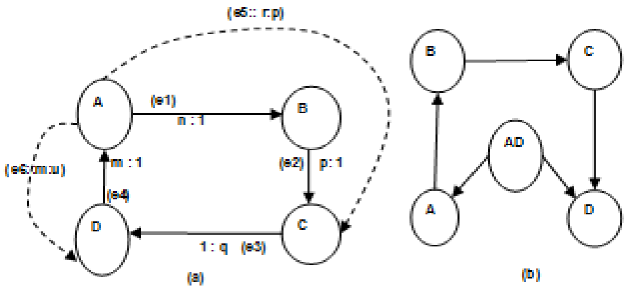


Figure 6: (a) Cyclic Reference and (b) Acyclic Reference

Theorem 3: *The reference graphs without self loop are always acyclic.*

We draw an edge from a node to another node if there is an n: 1 or 1: 1 mapping. In Fig. 6(a) we have the following mappings: (A<n: 1>B<m: 1>C<p: 1>D<q: 1>A)

To each B-vale, many A-values are associated (e1), to each C-value; many B-values are associated (e2). So, to one C-value, (n× p = r) A-values are associated (edge e5). Similarly, to each D-value, many C-values are associated (e3). By edge e4 we will find that to each A-value, many D-values are associated. But if we combine the edges e5 and e3 we will get the edge e6 meaning that, for each D value u=r× q values are associated. Therefore we can say that between D and A the mapping is many-to-many (u: m). But for many-to-many relationship we create a separate node with an edge to A and an edge to D, resulting a tree. Hence the graph cannot be cyclic and we get a reference graph as shown in Fig.6 (b).

IV. OUTLINE OF THE APPROACH

The proposed approach comprises of seven steps (see Fig. 7). First, identify the entities and their attributes in the system. Second step is to identify the key attributes of the set. In the third step, create the sets for the entity sets and add the key attributes in the key sets of the set. Fourth, identify the relationships among the entity sets and draw the intermediate graph showing the cardinalities of the relationships. In the fifth step convert the undirected edge of the intermediate graph by following the steps described in section **V**. Draw the reference graph and develop the abstract model. Then, define K_{SYS} and represent the reference graph as a matrix. Finally develop the mathematical model (DRS) for the database schema.

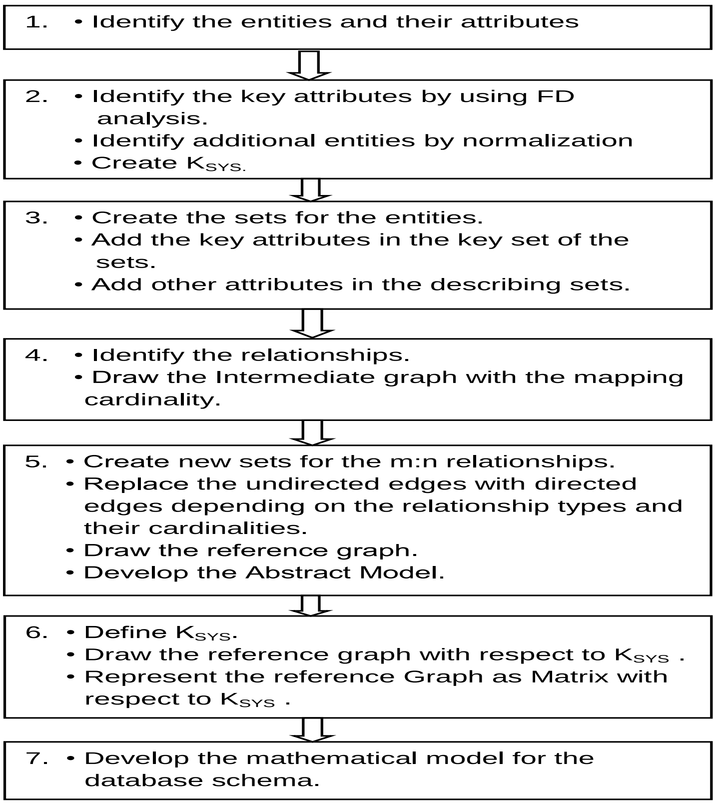


Figure 7: Outline of the Approach

V. STEPS IN DEVELOPING THE ABSTRACT MODEL

Step 1: Draw a node for each entity set.
Step 2: Find out the relationships among the sets (nodes) and join them with an undirected edge mentioning the mapping cardinalities. [As shown in Fig.9. Up to this step, we get the intermediate graph only.]
Step 3: Replace the edges with n: 1 or 1: n and 1:1 cardinality with directed edge from n-side to 1-side and label the edge as shown in Fig.10 ($k_{\text{NODE-NAME}}$). For an m: n edge or an m:n:p relationship, create a new set with an appropriate name , remove the undirected edge from the graph and draw directed edges from the new node to the nodes participating in the relationship. Label the edges accordingly. Define the key set for the newly created node.
Step 4: Develop an abstract mathematical model for the system as shown in Fig.11.

VI. REPRESENTATION OF THE REFERENCE GRAPH

The nodes of the reference graph can be defined as sets. Each set has a set of identifying attributes and a set of describing attributes. The reference graph of E-R model can be represented as a matrix (other representation is also possible, like sparse matrix). The contents of the matrix are the attribute(s) through which the vertices are associated. For example, let us consider the reference graph of Fig.-2(a) and Fig. 2(b). The reference graphs in Fig.2 (a) and Fig.2 (b) are represented in the matrices given in *Table-I* and *Table-II* respectively as follows.

In case of the reference are through composite attributes then the contents of the matrix are lists of attributes names.

TABLE-I : Reference Matrix for Fig. 2(b)

	STUDENT	BRANCH
STUDENT	X	BR-CODE
BRANCH	X	X

TABLE II: Reference Matrix for Fig. 2(c)

	SUBJECT	STUDENT	SCORE
SUBJECT	X	X	X
STUDENT	X	X	X
SCORE	ROLLNO	SUBCODE	X

VII. GENERATING DATABASE REQUIREMENT SPECIFICATIONS (DRS)

Step-1: Create the set K_{SYS}. Create a set for each entity in the E-R diagram. Add the identifying attributes in the key set of the set created for the entity and other attributes in the describing set. Add the primary key of the set to $K_{SYS.}$

Step 2: Search the reference graph to identify the nodes which have outdoing edges. Create a set for each node identified, if it not created in Step1.

Step 3: For each node identified in Step 2, add the attribute(s) specified in the outgoing edge labels of this node in the appropriate set (key set **or** describing set).

By following the above steps for the reference graph in figure Fig.3 (b), we get specifications of the database for the system described by the reference graph as follows.

By Step 1 we get the following sets.

K_{SYS} ={{Roll-No},{Sub-Code}}
STUDENT= {{{Roll-No}}, {Name, Address}}
SUBJECT = {{{Sub-Code}}, {Sub-Name, Tot-marks}}

By Step 2 and Step 3, we get the following set.

SCORE = {{Roll-No, Sub-Code}, {Sub-Name, Tot-marks}
K_{SYS} ={{Roll-No},{Sub-Code}, {Roll-No,Sub-Code}}

The complete database specifications are as follows.

DRS:STUDENT-SUBJECT
K_{SYS} ={{Roll-No},{Sub-Code}, {Roll-No,Sub-Code}}
SET STUDENT = {{{Roll-No}}, {Name, Address}}
SET SUBJECT = {{{Sub-Code}}, {Sub-Name, Tot-marks}}
SET SCORE = {{Roll-No, Sub-Code}, {Sub-Name, Tot-marks}

VIII. AN EXPLANATORY EXAMPLE

As an example, we are taking the E-R diagram for a Consultancy firm represented in KORTH's notations *[2]*. Fig.9 shows the intermediate diagram to get the reference graph in Fig.10. Fig. 11 shows the abstract mathematical model for the system. Table-III shows the matrix representation of the reference graph and finally from this, the mathematical model of the database schema is obtained as shown in Fig.13. This mathematical model is with respect to K_{SYS}.

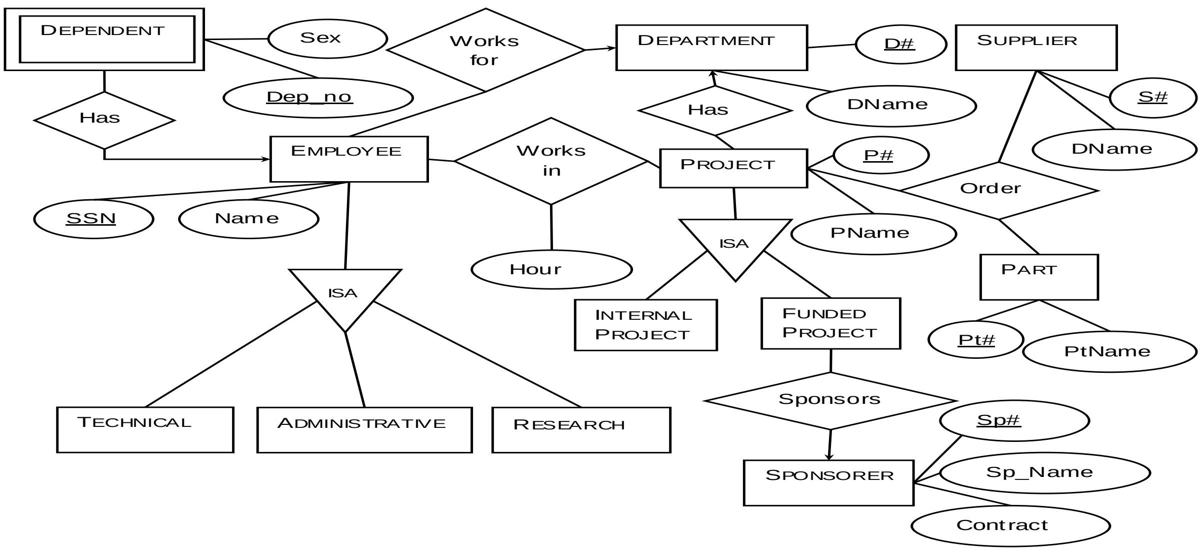
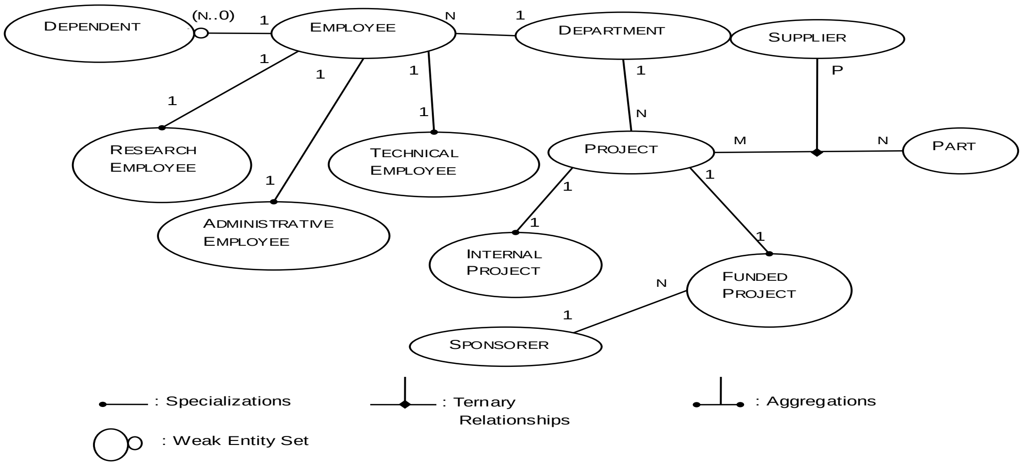


Figure 8: E-R model for a consultancy firm (KORTH'S NOTATION [2])



[Note that the intermediate graph can also be drawn directly by analysing the context without drawing an E-R diagram]

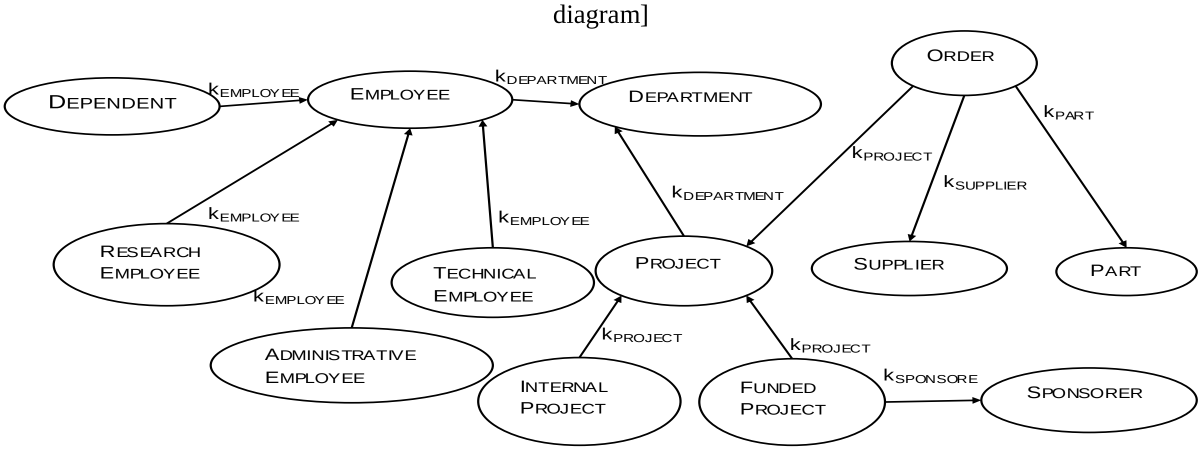


Figure10: Reference Graph for the System

From the reference graph in Fig.10, we will get the abstract mathematical model for the database of the system as shown in Fig.11. At this point, the decision of choosing the primary keys is pending until the actual mathematical model is developed.

ABSTRACT MODEL :: CONSULTANCY FIRM
SET EMPLOYEE= { {S-SN}, {{$k_{DEPARTMENT}$}, NAME} }
SET DEPARTMENT = { {D#}, {D_NAME} }
SET SUPPLIER= { {S#}, {SNAME} }
SET PART= { {PT#}, {NAME} }
SET SPONSORER= {{{SP#}, {ACC#}}, {SP_NAME, CONTRACT}}
SET PROJECT = { {P#}, {{$k_{DEPARTMENT}$}, NAME} }
SET DEPENDENT= { {$k_{EMPLOYEE}$}, DEP-NO}, {SEX} }
SET RESEARCH_EMPLOYEE= { {{$k_{EMPLOYEE}$}, {*}} , {RESEARCH_AREA} }
SET TECHNICAL_EMPLOYEE= { {{$k_{EMPLOYEE}$}, {*}}, {TECHNICAL_QUALIFICATION} }
SET ADMINISTRATIVE_EMPLOYEE = {{{$k_{EMPLOYEE}$}, {*}}, {PROJECT_SUPERVISED} }
SET INTERNAL_PROJECT = { {{$k_{PROJECT}$}, {*}}, {PNAME} }
SET FUNDED_PROJECT = { {{$k_{PROJECT}$}, {*}}, {{$k_{SPONSORER}$}, PNAME} }
SET ORDER= { {{ORDER#}, { {$k_{SUPPLIER}$}, {$k_{PROJECT}$}, {k_{PART}}} } }, {ORDER_DATE} }
[{} : set of other differentiating attributes]*

Figure 11: Abstract Mathematical Model for the System

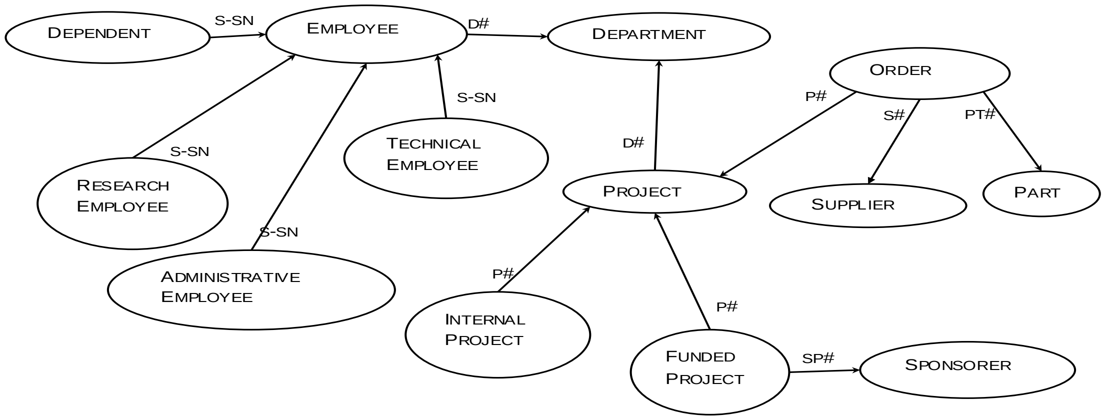


Figure 12: Reference Graph for the Database Schema for Consultancy Firm with respect to K_{SYS}.

Now, we get the reference graph for the database schema with respect to $\mathbf{K_{SYS}}$ from this model as shown in Fig.12. The set $\mathbf{K_{SYS}}$ contains the primary keys of the tables in the system. This is the final schema from which database tables are created.

TABLE III: Reference Matrix with respect to K_{SYS} for the Database of the Consultancy Firm

The DRS for the system is as follows:

DRS NAME= CONSULTANCY FIRM

TO / FROM	EMPLOYEE	DEPENDENT	DEPARTMENT	RESEARCH_EMPLOYEE	TECHNICAL_EMPLOYEE	ADMINISTRATIVE_EMPLOYEE	PROJECT	INTERNAL_PROJECT	FUNDED_PROJECT	ORDER	SUPPLIER	PART	SPONSORER
EMPLOYEE			S-SN										
DEPENDENT	S-SN												
DEPARTMENT													
RESEARCH_EMPLOYEE	S-SN												
TECHNICAL_EMPLOYEE	S-SN												
ADMINISTRATIVE_EMPLOYEE	S-SN												
PROJECT		D#											
INTERNAL_PROJECT							P#						
FUNDED_PROJECT							P#						SP#
ORDER							P#				S#	PT#	
SUPPLIER													
PART													
SPONSORER													

SET K={{S-SN},{S-SN,DEP-NO},{D#},{P#},{ORDER#},{PT#},{SP#}}

SET EMPLOYEE={{{S-SN}},{NAME}}

SET DEPENDENT= {{{(S-SN, DEP-NO)}}, {SEX}}

SET DEPARTMENT = {{{D#}}, {D_NAME}}

SET RESEARCH_EMPLOYEE= {{{S-SN}}, {RESEARCH_AREA}}

SET TECHNICAL_EMPLOYEE= {{{S-SN}}, {TECHNICAL_QUALIFICATION}}

SET SECRETARY={{{S-SN}} , {PROJECT_SUPERVISED}}

SET PROJECT = {{{P#}}, {NAME}}
SET INTERNAL_PROJECT = {{{P#}}, {PNAME}}
SET FUNDED_PROJECT = {{{P#}}, {PNAME}}
SET SUPPLIER={{{S#}},{SNAME}}
SET PART={{{PT#}},{NAME}}
SET ORDER={{{ORDER#},{ S#,P#, PT#}}, {S#,P#, PT#, ORDER_DATE}}
SET SPONSORER={{{SP#}}, { ACC#,SPNAME, CONTRACT}}

Figure-13: Mathematical Model for the Database Schema
[Note that these sets will be database tables and the reference graph can be transformed to schema diagram]

IX. RESTRUCTURING UNSTRUCTURED DATABASES

(A reverse engineering approach)

Still there are organizations which maintain data in an unstructured manner having redundancies. These databases can be restructured by using our approach in a reverse way. To remove these redundancies and to restructure the databases, we follow a reverse engineering approach, where we create sets for each of the database tables in the system. By equation 1, we know that there is reference from a set A to another set B if $A \cap B = \{bi\}$ $1 \leq i \leq m$ and ($bi \in B$) & ($b_i \in K_B$). If $\{A \cap B\}$ contains elements which do not belong to the key set of B but from the describing set of B, then there are redundancies. These redundancies can be removed by removing these elements, which are not in the key set, from one of the sets. If more than one attribute sets from the key set K_B of the set B is in $\{A \cap B\}$, then only one from$\{A \cap B\}$ must be kept in the set A, others should be removed from the set A, because it is impossible to have multiple references from A to B.

As an example, let us consider the following database tables with lots of redundancies.
STUDENT(ROLLNO, SNAME, ADDRESS,DOB,BLOOD_GROUP)
SUBJECT(SUBCODE, SUBNAME, TOTAL-MARKS, CREDITS)
MARKSDETAILS(ROLLNO, SNAME, SUBCODE, SUBNAME,MARKS-OBTAINED)
Now, we will represent these tables in our notations as follows
SET STUDENT={{{ROLLNO}}, {SNAME, ADDRESS,DOB,BLOOD_GROUP}}
SET SUBJECT={{{SUBCODE}}, {SUBNAME, TOTAL-MARKS, CREDITS}}
SETMARKSDETAILS={{{(ROLLNO,SUBCODE}},{SNAME,SUBNAME,MARKS-OBTAINED}}
From this, we get,
STUDENT∩SUBJECT=φ --------- (A)
STUDENT ∩ MARKSDETAILS={ROLLNO,SNAME} --------- (B)
MARKSDETAILS ∩ STUDENT ={ROLLNO,SNAME} --------- (C)
MARKSDETAILS ∩ SUBJECT ={SUBCODE,SUBNAME} --------- (D)

From B we get that there is no reference from STUDENT TO MARKSDETAILS because ROLLNO and SNAME both does not belong to the key set of MARKSDETAILS.(Note that ROlLLNO is not a key in MARKSDETAILS as ROLLNO and {ROLLNO} is different.). Similarly, we get that there is no reference from SUBJECT to MARKSDETAILS. But there are references from MARKSDETAILS to STUDENT and SUBJECT because ROLLNO belong to the key set of STUDENT and SUBCODE belong to the key set of SUBJECT. In (C) and (D), we see that the intersections include attributes from the describing sets. For omission of attributes from the set to remove redundancies we consider the intersections which produce references. Here we omit SNAME and SUBCODE from MARKSDETAILS and we get the following sets. The corresponding E-R model will be similar to the Fig. 2(b). The restructured database is as follows.

SET STUDENT={{{ROLLNO}}, {SNAME, ADDRESS,DOB,BLOOD_GROUP}}
SET SUBJECT={{{SUBCODE}}, {SUBNAME, TOTAL-MARKS, CREDITS}}
SET MARKSDETAILS={{{ROLLNO,SUBCODE}},{MARKS-OBTAINED}}

X. CONCLUSION

Though the E-R models look like graphs, we cannot directly apply the graph traversal algorithms to explore the database table references. We can only find out the relationships among the entities. By using the reference graph, we can easily explore the references among the database tables because the reference graphs can easily be represented by using common data structures. So the process of database design can be automated with less effort. By our approach the DRS i.e. the abstract database schema can be created by using the reference matrix and the sets defined at the time of analysing the context system. In this paper, we have discussed a new approach for data modelling. We have also discussed a method for generating the database specifications which can be used as a design document in the process of software development along with the Software Requirement Specification (SRS) Document. We are giving a generalized format for the database schema and can be used in any DBMS.

REFERENCES

[1]. P.P Chen, "Entity-Relationship Approach to Information Modelling and Analysis", Elsevier Science Publishers B.V. (North-Holand),1983, pp. 19-28

[2]. Il-Yeol Song , Mary Evans, E.K. Park "A Comparative Analysis of Entity-Relationship Diagrams", Journal of Computer and Software Engineering, Vol. 3, No.4 (1995), pp. 427-459

[3]. P. S. Dhabi et.al "Articulated Entity Relationship (AER) Diagram for Complete Automation of Relational Database Normalization", International Journal of Database Management Systems (IJDMS),Vol.2,No.2, May 2010, pp.84-100

[4]. David Heckerman, Chris Meek, Daphne Keller, "Probabilistic Entity-Relationship Models, PRMs, and Plate Models ", Proceedings of the 21 st International Conference on Machine Learning, Banff, Canada, 2004.pp.201-238

[5]. Anirban Sarkar "Conceptual Level Design of Semi-structured Database System: Graph-semantic Based Approach" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol.2, No.10, 2011, pp.112-121

[6]. Yan Yang, "Literature recommendation based on reference graph ", International Conference on Advanced Computer Theory and Engineering (ICATCTE), Vol.-3, 2010, pp. 400-404

[7]. Abraham Silberchatz, Henry F. Korth ,S.Sudarshan"Database System Concepts", McGraw-Hill Higher Education, 4th Edition,2002

[8]. Mark Graves, Ellen R. Bergeman, Charles B. Lawrence , "Graph Database Systems for Genomics, IEEE Engineering in Medicine and Biology special issue on Managing Data for the Human Genome Project, Volume: 14 , Issue: 6 1995,pp. 737 - 745

[9]. Madjid Tavana, Prafulla Joglekar, Michael A. Redmond "An Automated Entity-Relationship Clustering Algorithm for Conceptual Database Design", 2007, Information Systems, Vol. 32, No. 5, pp. 773-792.

[10]. Dmitri V. Kalashnikov and Sharad Mehrotr "Domain-Independent Data Cleaning via Analysis of Entity-Relationship Graph" ACM Transactions on Database Systems, Vol. 31, No. 2, June 2006, pp. 716–767.

[11]. Lisa Simasatitkul, Taratip Suwannasart, "A Tool for Generating Relational Database Schema from EER Diagram"Proceedings of International MultiConference of Engineers and Computer Scientists 2012,Vol-I, IMECS 2012, March 14-16, 2012, Hong Kong

[12]. Abhijit Sanyal, Anirban Sarkar, Sankhayan Choudhury, "Automating Web Data Model: Conceptual Design to Logical Representation", 19th Intl. Conf. on Software Engineering and Data Engineering (SEDE 2010), pp.94–99