An Efficient and Grain Preservation Mapping Algorithm: From ER Diagram to Multidimensional Model

Conference Paper · January 2005

DOI: 10.1007/11533962_30 · Source: DBLP

2 authors:

Yen-Ting Chen
National Chin-Yi University of Technology
7 PUBLICATIONS **104** CITATIONS

SEE PROFILE

Ping-Yu Hsu
National Central University
116 PUBLICATIONS **752** CITATIONS

SEE PROFILE

An Efficient and Grain Preservation Mapping Algorithm: From ER Diagram to Multidimensional Model

Yen-Ting Chen[1,2] and Ping-Yu Hsu[3]

[1] Department of Business Administration, National Central University, Jhongli City, Taoyuan 320, Taiwan, R.O.C.
[2] Department of Information Management, Lunghwa University of Science and Technology, Taoyuan 333, Taiwan, R.O.C.
s1441001@cc.ncu.edu.tw
[3] Department of Business Administration, National Central University, Jhongli City, Taoyuan 320, Taiwan, R.O.C.
pyhsu@mgt.ncu.edu.tw

Abstract. Many practitioners and researchers advocate that the designs of the data models of the data warehouses should incorporate the source data as much as possible to answer the finest levels of queries. On the other hand, the source data are very likely to come from systems designed with ER Diagrams. Therefore, many researches have been devoted to design methodologies to build multidimensional model based on corresponding source ER diagrams. However, to the best of our knowledge, no algorithm has been proposed to systematically translates an entire ER Diagram into a multidimensional model with hierarchical snowflake structures. The algorithm proposed in the paper promised to do so with two characteristics, namely, grain preservation and minimal distance from each table to the fact table. Grain preservation characteristic guarantees that translated multidimensional model has cohesive granularity among entities. The minimal distance characteristics guarantees that if an entity can be connected to the fact table in the derived model with more than one paths, the one with the shortest hops will always be chosen. The first characteristic is achieved by translating problematic relationships between entities with weight_factor attributes in bridging tables and enhancing fact tables with unique primary keys. The second characteristic is achieved by including a revised shortest path algorithm in the translating algorithm with the distance being calculated as the number of relationships required between entities.

1 Introduction

As enterprizes worldwide strive to compete on real time management, providing valuable information in time to managers to help them perform timely decisions has become a critical mission for MIS departments in organizations worldwide. Among the related applications installed, data warehouses play a vital role for integrating, storing, and querying data. The data collected in data warehouses come from various transactional processing systems, such as, ERP, POS, etc. The

F.F. Ramos et al. (Eds.): ISSADS 2005, LNCS 3563, pp. 331–346, 2005.

collected data are cleaned, integrated and organized into structures designed for easy access and quick comprehension, for the purpose of decision making [11, 12]. With its wide applications, many researches have been devoted to the study of proper models of data presentation [1–5, 7–10, 12, 13, 15–21]. The dominant data model for representing a data warehouse is multidimensional modelling[1], also known as a *malposition* model, which is composed of a fact table in the center and a set of dimensional tables in the peripheral. The fact table stores measures of performance indicators which managers are interested to know and dimensional tables provide the viewpoints or entry points to view the data. A fact table and corresponding dimensional tables are linked by storing the primary keys of dimensional tables in the fact table. Figure 1 is excerpted from [13] and shows a sample multidimensional model with a fact table and three-dimensional tables. The grain is the level of detail at which measurements or events are stored [13, 20]. To have cohesive querying results, the multidimensional models designed must have consistent grains, as pointed out by Inmon & Kimball [11, 13]. After the grain is declared, all measures in the fact table and all dimension tables must adhere to the grain. Otherwise, unexpected query result may be returned. Following is an example of the grain mismatch. In a typical supermarket visit, customers may buy several products in one transaction and the dollar amount of each product is aggregated into a total. Given a multidimensional model recording the total dollar amount of each transaction, the grain is in the level of transaction. (shown in Figure 2). If products are also stored as a dimension in the model, then querying the transaction amounts from product dimension will return figures that not only include amount of the product but also the amounts of other products purchased in the same transaction. The difference is due to the grain mismatch in the fact and the dimension table. The grain of the fact table is in transaction level, which may contains more than one products. Therefore, designing a model with coherent facts and dimensions is vital in designing multidimensional models.

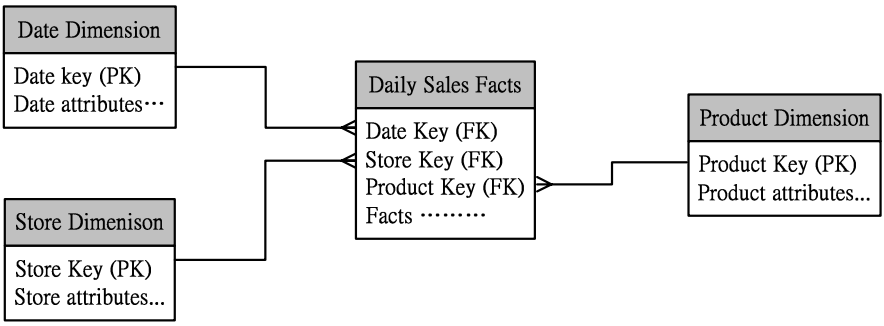

Fig. 1. A Sample Multidimensional Model; source [13]

[1] Multidimensional model is also referred as dimensional model in some studies

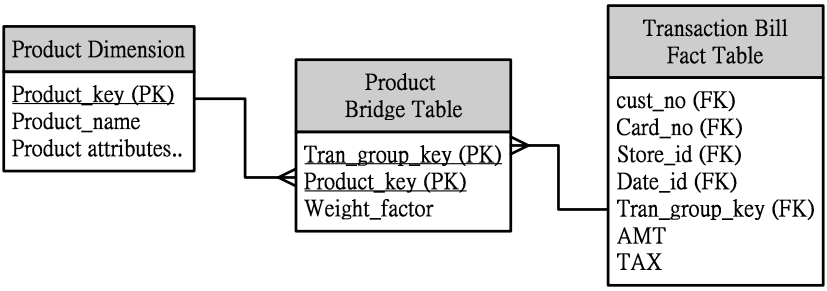

Fig. 2. The many-to-many relationship between the fact and dimension tables

On the other hand, data in a data warehouse are populated from source systems whose data probably are modeled by ER diagrams [6]. Therefore, the source ER diagrams show the most finest granular data that can be stored in a corresponding multidimensional model. If a systematic approach can be found to build multidimensional models to store data retrieved from the source systems modeled by ER diagrams, the work of model designers can be greatly reduced and more importantly, the errors committed can also be dramatically reduced.

Therefore, practitioners and researches have strived to propose methodologies to design multidimensional models from existing ER diagrams [3, 5, 8, 9, 14–16, 19–21]. Even though [12, 16, 20] acknowledged the importance of grain consistence, they provide only vague guidelines for the mapping and do not specify any concrete algorithms to perform task. Moody and Kortink [16] proposed a three-step method,including classifying entities, identifying hierarchies and producing dimensional models, for developing a dimension model from entity relational models. They also presented five optional schemas in the paper, from simple flat schema to complex snowflake schema. Song et al.[20] presented five methods to handle many-to-many relationships from an ERD to a dimension model. The possible solutions can be either adding a bridge table, denormalizing the dimensional table by positional-flag attributes or non-positional attribute, lowering the grain of the fact table to the dimension grain level, or lowering the grain of the fact table by separating data from the fact table.Bonifati et al. [4] presented a method to design a data mart. The method consists of three steps: top-down requirement analysis to elicit and consolidate user-requirements, bottom-up data model extract to form candidate data mart, and consolidation to derive the ideal data marts. Similarly, Cabibbo and Torlone [5] proposed method to obtain a Multidimensional schema from an underlying operational databases. The schema consists of a finite set of dimensions, a finite set of F-tables, and a finite set of level descriptions of the dimensions. Golfarelli et al. [8] presented a graphical conceptual model (Dimensional Fact model) for data warehouses and a semi-automated methodology to construct a tree-structured fact schema from an Entity-Relation schema. Marotta et al. [15] provided a set of transformation rules to trace the mapping between source logical schema and data warehouse logical schema. Tryfona et al., [21] presented a new model, the starER model, to

make semantics richer than traditional multidimensional model to record many-to-many relationships between fact and dimensional tables.Boehnlein et al. [3] proposed the SERM model to visualize existence dependencies between data object types.

The approach proposed in the paper differs from the other approaches in following ways:

(a) the algorithm systematically performs ER Diagrams to multidimensional model translation, given the fact table identified,
(b) the algorithm guarantees that adding a new entity to the structure does not change the grain of existing entities,
(c) the snowflake structure proposed by the algorithm takes the fewest relationship to connect dimensions and the given fact table.

The remainder of this paper is organized as follows:

Section 2 formally defines grain preservations. Section 3 presents the ER to multidimensional translation algorithm. Section 4 explains how the two characteristics are achieved by the algorithm.

Section 5 uses a case to demonstrate the algorithm. Finally, the summary and future work is presented in Section 6.

2 Grain Preservation

As Kimball pointed out [13], a multidimensional model in general contains a fact table and a set of dimension tables. Each dimension table has a primary key which is also a foreign key of the fact table. The primary key of the fact table is the composition of all the foreign keys stored in the dimension tables.

If grain mismatch happens between the fact table and any of the dimension tables then the query result may be wrong [13, 20]. The erroneous queries return values that aggregate individual measures more than one time. Hence, a multidimensional model with the consistent grain should aggregate at most one copy of individual measure, regardless of the dimensions users querying along. Before formally defining grain preservation, we need to define two operators, namely, $\sum$ and $\star$.

Definition 1 *Given a table, T, with m of its attributes are measures,namely, $a_{i1}, \ldots, a_{im}$, and n of its attributes are weight factors, namely, $a_{j_1}, \ldots, a_{j_n}$.*

- $\sum a_{ik} = \sum_{t \in T} t.a_{ik}$, *for* $1 \leq k \leq m$
- $\sum(T) = \langle \sum a_{i1}, \ldots, \sum a_{im} \rangle$.
- $\star(T) = \{t' | \forall a \in$ *attriubtes of* $T, t \in T, (a \notin \{a_{i1}, \ldots, a_{im}\} \rightarrow t'.a = t.a) \wedge (a \in \{a_{i1}, \ldots, a_{im}\} \rightarrow t'.a = t.a * t.a_{j_1} * \ldots * t.a_{j_n})\}$

With the $\star$ operator, a measure can be refined to finer grain. Table 1 shows a sample table with customer# as a non measure attribute, amount and cost as measures and weight_factor$_1$, weight_factor$_2$ as weight factor attributes.

Table 2 shows the result of applying $\star$ operator on Table 1.

Table 1. A sample Table

customer#	amount	cost	weight_factor$_1$	weight_factor$_2$
c125	50	30	0.4	0.2
c125	50	30	0.6	0.8
c125	50	30	0.4	0.8
c125	50	30	0.6	0.2
c127	40	30	1.0	1.0

Table 2. Applying $\star$ to Table 1

customer#	amount	cost	weight_factor$_1$	weight_factor$_2$
c125	4	2.4	0.4	0.2
c125	24	14.4	0.6	0.8
c125	16	9.6	0.4	0.8
c125	6	3.6	0.6	0.2
c127	40	30	1.0	1.0

In a multidimensional model with snowflakes, a table can be added as a dimension table, which connects to the fact table directly, or as a table in the snowflake hierarchy, which is composed by a set of tables connected as a tree with the fact table as the root. A table added to a multidimensional model without breaking the existing grain provides an entry point to correctly summary measures.

Definition 2 *Given a multidimensional model with a fact table, F, and a table T, if adding T to the multidimensional model results in a path of $F, D_1, \ldots, D_k, T$ connecting T to F, and $\sum(\star(F \bowtie D_1 \ldots \bowtie D_k \bowtie T)) = \sum(F)$, where $\bowtie$ is a natural join operator then the addition is called* Grain Preservation.

3 The Mapping Algorithm

The section shows a relationship translation algorithm, which translates entities and relationships in an ER diagram to dimension tables in a multidimensional model while keeping grain preservations and taking the least join operators.

Given the ER diagram of a source system, the entities connecting to others with some many-to-many relationships and including additive numeric attributes are candidates for being the fact tables [12]. The paper assumes that a table in the source ER diagram is identified as the fact table which contains several fact attributes, also known as measures.

Given a source ER diagram with $\langle \boldsymbol{E}, \boldsymbol{R} \rangle$, where $\boldsymbol{E}$ is the set of entities and $\boldsymbol{R}$ is the partial functions of relationships in the ER Diagram. $\boldsymbol{R} : \boldsymbol{E} \times \boldsymbol{E} \rightarrow \{$'1-1','1-M','M-1','M-N'$\}$, where '1-1','1-M', 'M-1' and 'M-N' denote the cardinality of the relationships.

A multidimensional model is a $\langle \boldsymbol{DE}, \boldsymbol{DR} \rangle$, where $\boldsymbol{DE}$ is the set of tables in the model, and $\boldsymbol{DR}$ is the partial function of relationships between the tables. Every entity in $\boldsymbol{E}$, and $\boldsymbol{DE}$ is assumed to have a primary key.

3.1 Naive Mapping Rules

The naive mapping rules analyze the $\boldsymbol{R}$ between entities in the source ER diagram and translate corresponding entities and relationships into multidimensional model. Translation of '1-M' and 'M-to-N' relationships may produce grain mismatch if they are not handled carefully. As pointed out in [12, 13, 20],the mismatch can be corrected by lowering the grain in the fact tables or the grain in the dimension tables. In the case of snowflaked multidimensional model, we argue that lowering the grain of dimension tables should be more preferable since the same methodology can be applied to lower the grains of tables down in the snowflake hierarchy.

If $\boldsymbol{R}(E_i, E_j)$ exists and $E_i \in \boldsymbol{DE}$ then E_j is added to the $\boldsymbol{DE}$ in following ways:

– Rule#1: $\boldsymbol{R}(E_i, E_j)$ = 'M-to-1'
In this case, the translation is straightforward.

$$\boldsymbol{DE} = \boldsymbol{DE} \bigcup \{E_j\}$$
$$\boldsymbol{DR} = \boldsymbol{DR} \bigcup \{\boldsymbol{DR}(E_i, E_j) = \boldsymbol{R}(E_i, E_j)\}$$

Figure 3 shows an example of such cases.

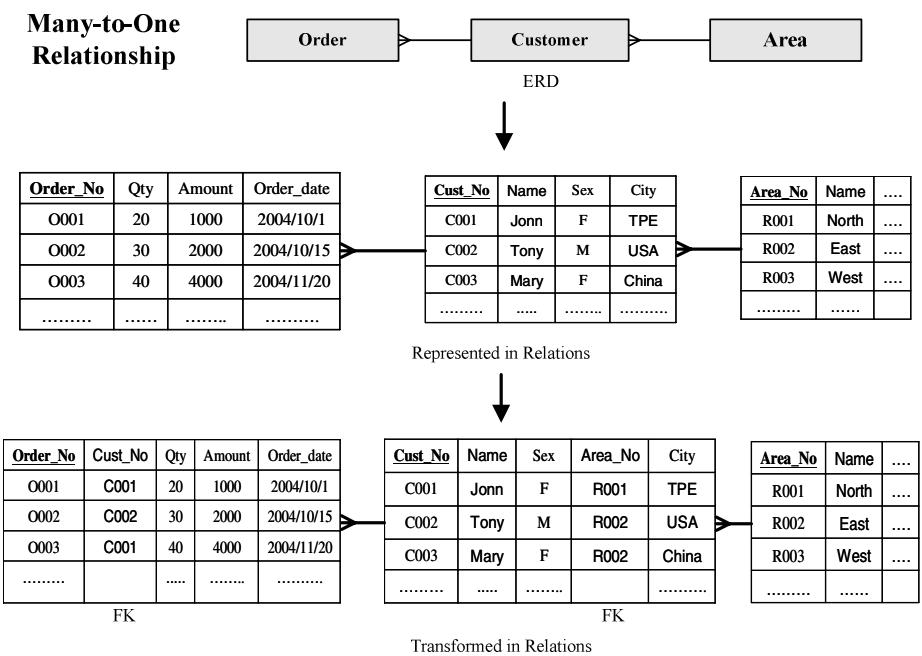

Fig. 3. Transformation in a 'many-to-one' relationship

– Rule#2: $\boldsymbol{R}(E_i, E_j)$ = '1-to-M'
Since the grain of E_j is finer than the grain of E_i, an attribute of weight factor is added to E_j to tune the grain of E_j.

$$\boldsymbol{E}'_j = \boldsymbol{E}_j + \text{weight\_factor}$$
$$\boldsymbol{DE} = \boldsymbol{DE} \bigcup \{E'_j\}$$
$$\boldsymbol{DR} = \boldsymbol{DR} \bigcup \{\boldsymbol{DR}(E_i, E'_j) = \boldsymbol{R}(E_i, E_j)\}$$

Figure 4 shows an example of '1-M' relationship translation.

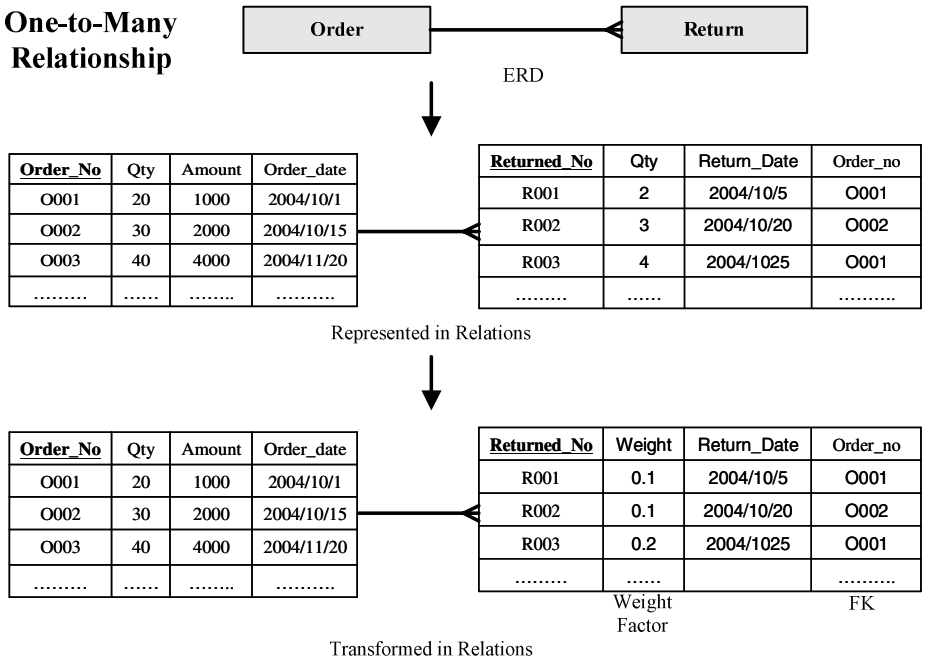

Fig. 4. Translation of a 'one-to-many' relationship

– Rule#3: $\boldsymbol{R}(E_i, E_j)$ = 'M-to-N'
Since The grains in the two tables are incompatible, a bridging table $\boldsymbol{B}$ is added to tune the grain. The table has two foreign keys coming from E_j, and E_i, respectively, and an attribute of weight_factor[12, 20]. The foreign key from E_i groups entries in E_j so that the combinations of the entries have the same grains as the corresponding entries in the E_i. The attribute of weight_factor records the contribution of the entries in the group. The summation of weight_factor in each group should be equal to one.

$$\boldsymbol{B} = \langle weight\_factor \rangle$$
$$\boldsymbol{DE} = \boldsymbol{DE} \bigcup \{E_j, \boldsymbol{B}\}$$

$$\boldsymbol{DR} = \boldsymbol{DR} \bigcup \{\boldsymbol{DR}(E_i, \boldsymbol{B}) = \text{`}1 - to - M'\}$$
$$\boldsymbol{DR} = \boldsymbol{DR} \bigcup \{\boldsymbol{DR}(\boldsymbol{B}, E_j) = \text{`}M - to - 1'\}$$

Figure 5 shows such an example. The algorithm of transformation rules is shown in Figure 6.

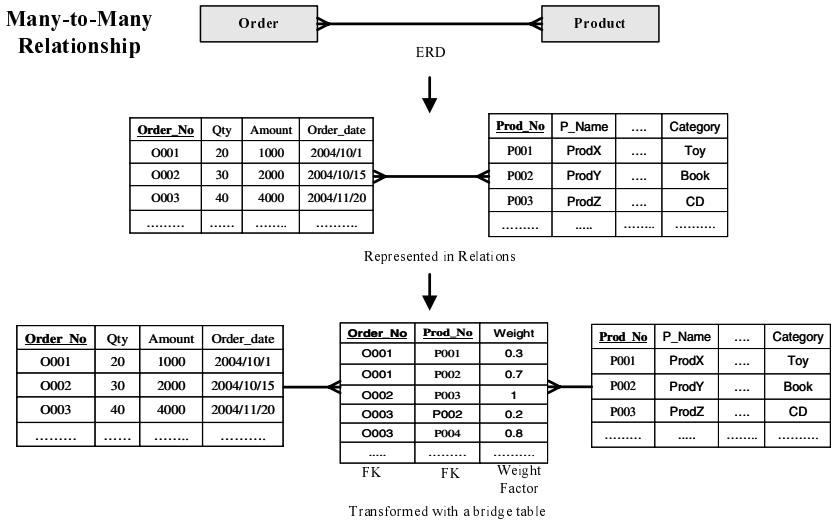

Fig. 5. Translation of a 'many-to-many' relationship

Algorithm: MR
Input: $\boldsymbol{E}_0$ //the fact table
$\langle \boldsymbol{E}, \boldsymbol{R} \rangle$ //the original ERD
Output: $\langle \boldsymbol{DE}, \boldsymbol{DR} \rangle$ //the desired multidimensional model

Begin
$\langle \boldsymbol{L}_0, \boldsymbol{L} \rangle$ = Shortest_distance($E_0, \langle \boldsymbol{E}, \boldsymbol{R} \rangle$)
$\boldsymbol{DE} = \{\boldsymbol{E}_0\}$
While $\exists E_i, E_j \in \boldsymbol{E}$
and $E_i \in \boldsymbol{DE} \wedge E_j \notin \boldsymbol{DE}$
and $\boldsymbol{L}(E_0, E_j) = \boldsymbol{L}(E_0, E_i) + min(\boldsymbol{L}_0(E_i, E_j), \boldsymbol{L}_0(E_j, E_i))$ do {
case $\boldsymbol{R}(E_i, E_j)$ {
when (1,1) then apply rule#1
when (M,1) then apply rule#1
when (1,M) then apply rule#2
when (M,N) then apply rule#3
}
}
Return $\langle \boldsymbol{DE}, \boldsymbol{DR} \rangle$
End;

Fig. 6. Relationship Translation Rules

3.2 Applying the Mapping Rules to an ER Diagram

This section further explains the process of building an entire multidimensional model from a given source ER diagram. The algorithm proposed assume that a fact table in the source ER diagram has been identified and strives to find a multidimensional model with the least number of number of relationships.

Since an ER Diagram can be translated to more than one multidimensional model, the major decisions lie on the selection of relationships to form the translated multidimensional model. The point is illustrated by following example.

Example 1 *An original ERD as shown in Figure 7(A) that includes several loop relationships. A loop among the snowflake tables causes problems in aggregating measures when querying data. A loop means there are more than one path to join a given table in the loop to the fact table. The different path may aggregate measures differently and cause confusion. Therefore, loop in the given ER diagram has to be broken when translating the ER diagram into a multidimensional model.*

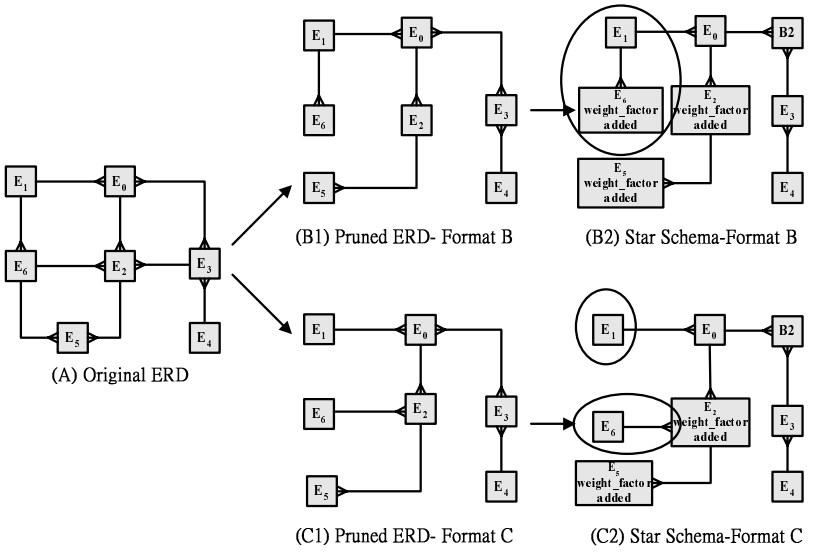

Fig. 7. An example of the same ERD being transformed into different multidimensional models

A loop can be broken in several ways. For example, the original ER diagram shown in 7(A) can be translated to two different ERDs (in Figure 7(B1)(C1)). Next, different multidimensional models (see Figure 7(B2)(C2)) are generated by applying the naive mapping rules shown in section 3.1. The differences between the two figures are highlighted in circles.

Hence, given an ER diagram, there are more than one way to build a corresponding multidimensional model. The paper proposes to use a shortest path

algorithm to derive the multidimensional model where each entity has the shortest path to the fact table. The distances between entities and the fact table are counted by the number of relationships between them, since the more the relationships, the more join operators are needed to perform queries. Readers will find that the calculation of distances can be changed to many other formula, such as the estimated numbers of tuples participated in each join, without jeopardizing the integrity of the algorithm.

To compute the shortest distance between the fact table and all other tables, an initial distance matrix has to be built. The initial matrix is formed by scanning the entire ER diagram and for each entity pair that are connected by relationships other than 'M-to-N', the initial value of 'one' is assigned. For relationships that are 'M-to-N', an initial values of 'two' are assigned to the corresponding entries since a bridging table will be needed in the translation. After deriving the initial matrix, the algorithm then calculates the shortest path matrix, which is asymmetric. The diagonal elements are filled with zeros.

Zeros in diagonal entries of the distance matrix represent that the distance between an entity and itself is zero. Such an assumption may in contradiction to entities with self reference relationship in ER Diagrams. However, in the paper, the translated multidimensional model is assumed to be free of self reference relationships. The assumption is based on the widely adopted practice that most self reference relationships are flatten into corresponding entities to save query processing time. The algorithm of the Shortest Distance Computing is shown in Figure 8.

4 Correctness Proof

The section proves that given an ER diagram of $\langle \boldsymbol{E}, \boldsymbol{R} \rangle$ with a fact table identified, the algorithm of MR returns a $\langle \boldsymbol{DE}, \boldsymbol{DR} \rangle$, which satisfies *Grain Preservation* in the process and all tables are connected to fact tables with the least relationships.

Theorem 1 *Given an ER diagram and a fact table, the process of adding entities to* $\langle \boldsymbol{DE}, \boldsymbol{DR} \rangle$ *by algorithm MR is* Grain Preservation.

Rationale
Only the addition of entities with relationships of '1-to-M' and 'M-to-N' may challenge grains of a multidimensional model. However, since the summary of weight_factors in each group in the two cases have to equal to one, the grain is still be kept according to Definition 2.

The fewer the relationships between entities and the fact table in a multidimensional model, the more efficient the queries issued from the entity can be processed.

Theorem 2 *Given an ER diagram and a fact table, the multidimensional model discovered by algorithm MR connecting each entity to the fact table with the least number of relationships.*

```
Algorithm: Shortest_Distance
Input:   E_0          //the fact table
         ⟨E, R⟩       //the original ERD
Output  L_0: Initial Distance Matrix
        L : Shortest Distance Matrix
        Begin
            Fill L_0 with ∞
            // scan the original ERD and assign initial value to each relationship
            for  i=0 to | E | do {
                for j=1 to | E | do {
                    for each R(E_i, E_j)  do {
                        if R(E_i, E_j) = 'M-to-1' then L_0(E_i, E_j) = 1
                        elseif E(E_i, E_j) = '1-to-1' then L_0(E_i, E_j) = 1
                        elseif R(E_i, E_j) = '1-to-M' then L_0(E_i, E_j) = 1
                        elseif R(E_i, E_j) = 'M-to-N' then L_0(E,E_j) = 2
                    }
                }
            }

            //Computing shortest path matrix
            for i = 0 to |E| {
                for j = 0 to |E| {
                    for k = 0 to |E| {
                        L(E_j, E_k) = min(L(E_j, E_k), L(E_j, E_i) + L(E_i, E_k))
                    }
                }
            }
            return L_0, L
        End;
```

Fig. 8. Computing the Shortest Distance Between Entities and the Fact Table

Rationale

The distances between entities to the fact table are computed by the relationships needed to connect the two tables. MR adds only one relationship to the multidimensional model when needed, which is the most efficient way found up to date [20] when keeping the grains of fact tables are mandatory. Besides, the MR algorithm uses the shortest path algorithm to find the paths with the least relationships to connect entities with the designated fact table.

5 A Sample Case

In order to show the translation process, a sample is demonstrated in the section. Figure 9(a) shows a sample ER Diagram derived from a commercial sales order tracking system. The designated fact table is the order table. With several loops in the ER diagrams, the diagram can be translated into more than one multidimensional model. The initial distance between connected entities are marked in

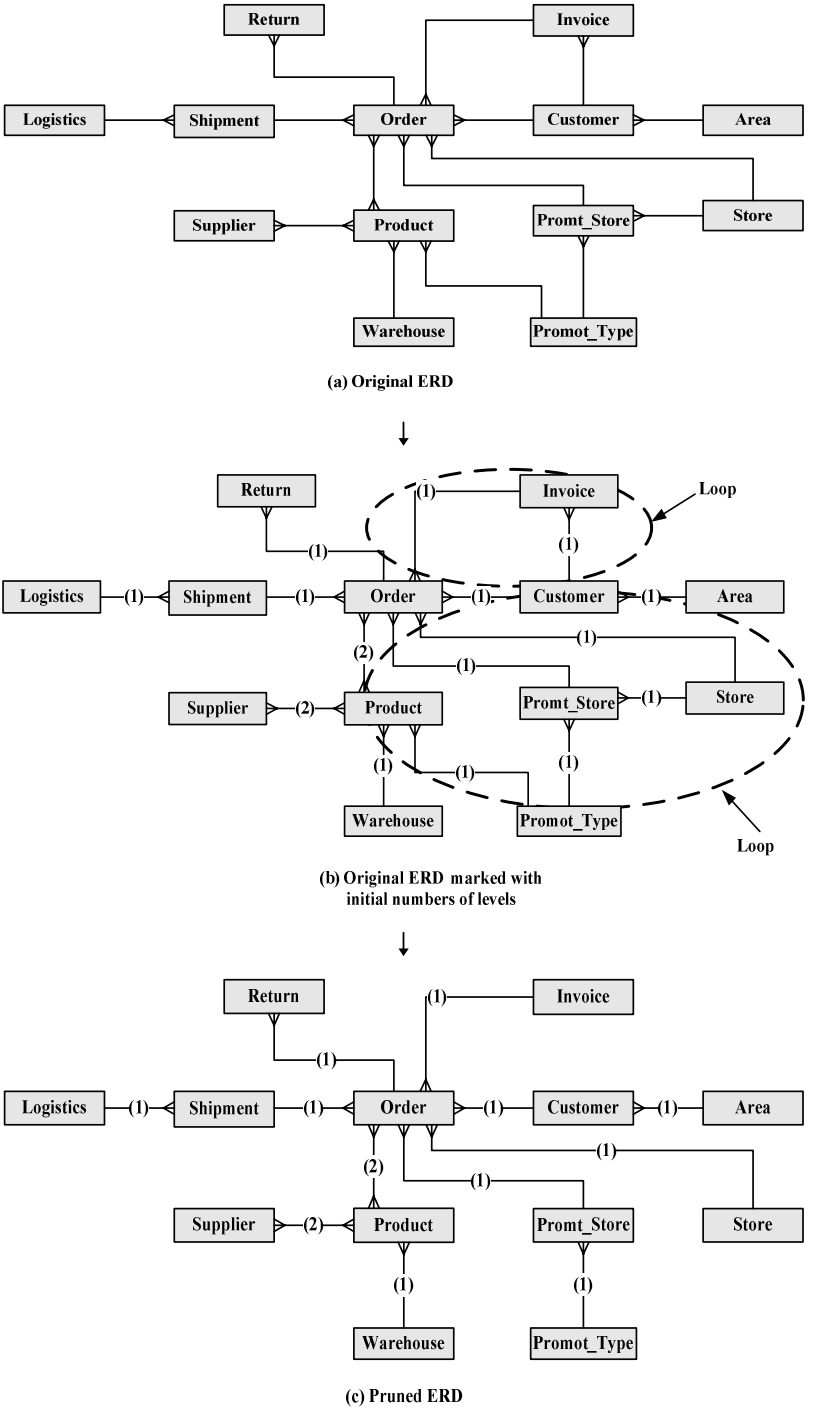

Fig. 9. Computing Distances between Dimensions and the Fact table

Figure 9(b). The calculated shortest distances between each entity and the fact table is shown in Figure 9(c).

The corresponding shortest distance matrix is shown in Table 3.

Table 3. the shortest distance matrix from entity Order to other entities

	Ordr	*Rtrn*	*Inv*	*Logs*	*Shp*	*Cust*	*Area*	*Sup*	*Prdt*	*PmtS*	*Str*	*WH*	*PmtT*
order	0	1	1	2	1	1	2	4	2	1	1	3	2
Return	∞	0	2	3	2	2	3	5	3	2	2	4	3
Invoice	∞	∞	0	3	2	1	2	6	4	3	3	5	4
Logisitcs	∞	∞	∞	0	1	3	4	6	4	3	3	5	4
Shipments	∞	∞	∞	∞	0	2	3	5	3	2	2	4	3
Customer	∞	∞	∞	∞	∞	0	1	5	3	2	2	4	3
Area	∞	∞	∞	∞	∞	∞	0	6	4	3	3	5	4
Supplier	∞	∞	∞	∞	∞	∞	∞	0	2	4	5	3	3
Product	∞	∞	∞	∞	∞	∞	∞	∞	0	3	3	1	1
Promt_Store	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	1	3	1
Store	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	4	2
Warehouse	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0	2
Promot_type	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0

The algorithm MR is then applied to translate selected relationships and corresponding entities into a multidimensional model. The transformation process is illustrated in the following steps and also shown in Figure 10:

(a) The designated fact entities *Order* is added into ***DE*** (identified as Step 1 in Figure 10).
(b) Based on the elements in the first row in the shortest distance matrix to transform relationships. Entity *Return, Invoice, Shipment, Customer, Promot_Store, Store*, and *Data warehouse* are candidates since their distances to Entity *Order* are the shortest. Assuming Entity *Return* is processed first. Since the relationship between *Return* and *Order* is one-to-many; a weight_factor attribute is added into *Return*. *Return* is added into ***DE*** and the ***R***(*Order*, *Return*) is added to ***DR*** (identified as Step 2 in Figure 10).
(c) With the addition of Entity *Return* to ***DE***, the eligible entities becomes Entity *Invoice, Shipment, Customer, Prmot_Store, Store*. Assuming *Invoice* is processed next. Since the relationship between *Invoice* and *Order* is many-to-one, the *Invoice* dimension in added into ***DE*** and the relationship of ***R***(*Order*, *Invoice*) is added to ***DR*** (identified as Step 3 in Figure 10).
(d) Entities *Shipment*, *Promot_Store*, *Customer* and *Store* are added to the model as in previous steps (identified as Step 4 through Step 7 in Figure 10).
(e) Entity *Product* is processed then, the relationship between *Product* and *Order* is many-to-many; a bridge table *Prod_B* with a weight_factor attribute will be added in between to tune the grain, the relationships of ***R***(*Order*, *Prod_B*) and ***R***(*Prod_B*, *Product*) are added to ***DR*** (identified as Step 8 in Figure 10).

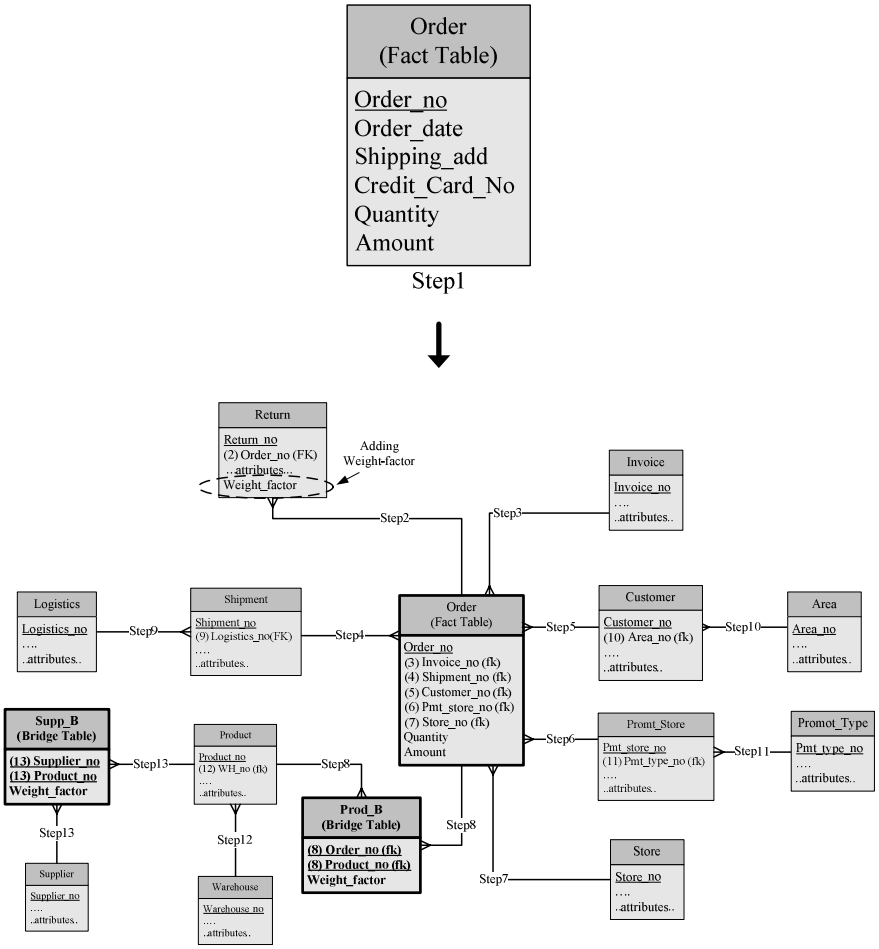

Fig. 10. The Translated multidimensional model $\langle \boldsymbol{DE}, \boldsymbol{DR} \rangle$

(f) With addition of all entities in level one, Entity *Logistics*, *Promot_type*, *Area*, *Warehouse* and *Supplier* are becoming eligible to be processed next. Entity *Logistics*, *Promot_type*, *Area*, *Warehouse* are process as entity *Invoice* done in Step 5 (identified as Step 9 through Step 12 in Figure 10).

(g) Finally, Entity *Supplier* is processed, the transformation process is same as Entity *Product*; thus, a bridge table *Supp_B* is added into $\boldsymbol{DE}$ and two relationships $\boldsymbol{R}(Product, Supp\_B)$ and $\boldsymbol{R}(Supp\_B, Supplier)$ are added to $\boldsymbol{DR}$ (identified as Step 13 in Figure 10).

6 Summary and Conclusions

As enterprizes place top priority on real time management, Data Warehouse systems have become critical information analytical tools. However, most com-

panies suffer from the lack of experienced Data Warehouse design professionals to effectively design multidimensional models. On the contrary, people with ERD concepts and experiences are far more widely available. Hence, deriving data warehouse schema from ER diagram may be one of the best way to create multidimensional models. However, a corporate Data Warehouse is not easy to be built by inexperienced Data Warehouse team in a short time. The techniques and mapping rules presented here will be valuable for them to have a quick and correct start. This paper presents a tool to derive data warehouse schema from ER diagrams. The transformation rules presented in this paper give Data Warehouse team a great tool to start with.

The main contribution of this paper providing an efficient and correct algorithm to translate ER diagrams into multidimensional model. The algorithm is efficient because every dimension table in the formed proposed hierarchy is connected to the fact table through the least expensive path. The algorithm is correct in that while adding new entities to existing multidimensional models, it still preserves the grain of the original dimensional model.

The issue of data warehouse model design should also include user requirement taking, verification and integration of the requirement with atomic grain data. Therefore, the work presented in this paper is just a foundation for further research of systematic data warehouse schema design.

Acknowledgement

The authors would like to acknowledge the financial support by the National Science Council, Taiwan, through the project no. NSC93-2416-H-008-010.

References

1. L. Baekgaard and F. Alle. Event-entity-relationship modelling in data warehouse environment. In *ACM Second International Workshop on Data Warehousing and OLAP (DOLAP)*, pages 9–14, Kansas City, Missouri, USA, November 6 1999. ACM.
2. P.A. Bernstein and E. Rahm. Data warehouse scenarios for model management. In *ER (2000) Conference Proceedings*, pages 1–15, Salt Lake City, Utah, USA, October 2000. Springer.
3. M. Boehnlein and A. Lbrich. Deriving initial data warehouse structures from the conceptual data models of the underlying operational information systems. In *ACM Second international Workshop on Data Warehousing and OLAP (DOLAP)*, pages 15–21, Kansas City, Missouri, USA, November 1999.
4. A. Bonifati, F. Cattaneo, S. Ceri, A. Fuggetta, and S. Paraboschi. Designing data marts for data warehouse. In *ACM Transactions on Software Engineering and Methodology*, volume 10 of *4*, pages 452–483, October 2001.
5. L. Cabibbo and R. Torlone. A logical approach to multidimensional databases. In *In Proceedings of the International Conference on Extending Data Base Technology*, pages 183–197, Balencia, Spain, March 1998.
6. P. P.S. Chen. The entity-relationship model -toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, March 1976.

7. E. Franconi and U. Sattler. A data warehouse conceptual data model for multidimensional aggregation. In *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'99)*, pages 13–1 13–10, Heideliberg, Germany, 1999.
8. M. Golfarelli, D. Maio, and S. Rizzi. Conceptual design of data warehouses from er schemes. In *Proceedings of the Hawaii International Conference On system Sciences*, Kona, Hawaii, January 1998.
9. M. Golfarelli and S. Rizzi. A methodological framework for data warehouse design. In *ACM First International Workshop on Data Warehousing and OLAP*, pages 3–9, Washington D.C., United States, November 1998.
10. B. Husemann, J. Lechtenborger, and G. Vossen. Conceptual data warehouse design. In *Proceedings of the International Workshop on Design and Management of Data Warehouse(DMDW2000)*, page 6, Stockholm, Sweden, 2000.
11. W.H. Inmon. *Building the Data Warehouse.* New York: John Wiley & Sons, Inc., third edition, April 2002.
12. R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite. *The Data Warehouse Life cycle Toolkit.* New York: John Wiley & Sons, Inc., 1998.
13. R. Kimball and M. Ross. *The Data Warehouse Toolkit.* New York: John Wiley & Sons, Inc., second edition, 2002.
14. T.M. Krippendorf and I.Y. Song. The translation of star schema into entity-relationship diagrams. In *Eighth International Conference and Workshop on Database and Expert-systems Applications (DEXA'97)*, pages 390–395, Toulouse, France, September 1997.
15. A. Marotta and R. Ruggia. Data warehouse design: A schema-transformation approach. In *Proceedings of the XXII International Conference of the Chilean Computer Science Society (SCCC 2002)*, pages 153–162. IEEE-CS, November 2002.
16. D. L. Moody and M. A.R. Kortink. From enterprise models to dimensional models: A methodology for data warehouse and data mart design. In *Proceedings of the International Workshop on Design and Management of Data Warehouse(DMDW2000)*, page 5, Stockholm, Sweden, 2000.
17. B. Pedersen and C.S. Jensen. Multidimensional data modeling for complex data. In *Proc. of 15th ICDE*, pages 336–345, Sydney, Austrialia, March 1999.
18. F. Ravat, O. Teste, and G. Zurfluh. Towards data warehouse design. In *Proceedings of the 1999 ACM CIKM International Conference on Information and Knowledge Management*, pages 359–366, Kansas City, Missouri, USA, November 1999.
19. C. Sapia, M. Blaschka, G. Hofling, and B. Dinter. Extending the e/r model for the multidimensional paradigm. In *Proceedings of the Workshops on Data Warehousing and Data Mining: Advances in Database Technologies, LNCS Vol. 1552*, pages 105–116. Springer-Verlag, 1998.
20. I.Y. Song, C. Medsker, W. Rowen, and E. Ewen. An analysis of many-to-many relationships between fact and dimension tables in dimensional modeling. In *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2001)*, pages 13–1 13–10, Interlaken, Switzerland, June,4 2001.
21. N. Tryfona, F. Busborg, and J.G.B. Christiansen. starer: A conceptual model for data warehouse design. In *ACM Second International Workshop on DataWarehousing and OLAP (DOLAP)*, pages 3–8, Kansas City, Missouri, USA, November 1999.