A Systematic Approach for Transformation of ER Schema to Dimensional Schema

Khurram Shahzad
Department of Computer &
Systems Sciences,
Royal Institute of Technology &
Stockholm University,
Forum 100, SE 164 40, Kista,
Stockholm, Sweden.
mks@dsv.su.se

Abid Sohail
Department of Computer Science,
COMSATS Institute of
Information Technology,
1-Km Defense Road, Off Raiwand
Road, Lahore, Pakistan.
abidbhutta@ciitlahore.edu.pk

ABSTRACT

Development of Star model for data warehouse (DW) requires ample amount of time, experience, expertise and adequate knowledge of data warehouse designing steps. As an alternative to data warehouse designing steps, transformation techniques are used. These techniques transform ER model (of relational OLTP) to Star model/s. In this study, we present and evaluate a systematic approach for transformation of ER model to Star model. Based on an experiment, we prove that, by using our approach, inexperienced designers can produce Star models identical to the star models produced by experienced data warehouse designers.

Keywords

Decision support systems, computer-based decision aids, Data Warehouse, ER model, Star model.

1. INTRODUCTION

Transactional systems, with their entity-relationship (ER) models, are unable to meet analytical requirements of an enterprise, because of the following reasons [1]: a) transactional systems don't store historical data, b) transactional systems are optimized for handling day to day transactions only and c) in transactional systems data is spread across a number of applications. Data warehouse, with its dimensional model, evolved as a solution because, a) DW stores historical data, b) DW provides integrated access to data, c) DW is optimized for meeting analytical requirements of an enterprise [8]. Dimensional schema of DW has three types, Star schema, Snowflake schema and Constellation schema. It is notable that, the terms schema and model are used alternatively in this paper.

According to Paulraj [1], star schema development process consists of the following phases: identification of business process, user requirements gathering, identification of grains, facts identification and dimensions definition. Accomplishing all these steps, for designing star schema, is not only complex and it requires ample amount of time but also entails adequate knowledge of data warehousing [5]. In addition to that, designing a star schema requires immense experience and expertise, due to which data warehouse designing experts are not easily available in market. An alternative approach to star schema development is, translation of Entity-Relationship model to Star model. Researchers have development semi-automated [7] and manual mechanisms [6, 8] (also known as transformation techniques) for transforming ER model to Star model.

In this study we aim to propose and evaluate a systematic approach for transformation of ER schema to star schema. In order to propose a new transformation approach, we first motivate the need of transformation and then identify shortcomings of existing approaches, which are found during application of these approaches on case studies. Keeping in view the shortcomings, we propose a systematic translation approach.

2. MOTIVATION AND TRANSFORMATION

Designing a dimensional model for data warehouse is a complex task due to a number of issues associated with dimensional modeling. Paulraj [1] & Kimball [5] have defined some steps for development of star schema but these steps not only require ample knowledge of all designing issues of dimensional modeling (DM) but also it requires a lot of data warehousing designing experience and expertise. Lack of considerations of DM designing issues may result in formulization of inconsistent dimensional model.

To achieve precision in modeling, we need a transformation technique that facilitate designer to translate an entity-relationship model into a dimensional model. In the absence of transformation method, different designers may end up producing different dimensional models from single ER model. Similarly,

FIT'09, December 16–18, 2009, CIIT, Abbottabad, Pakistan.
Copyright 2009 ACM 978-1-60558-642-7/09/12....$10.

inconsistencies may arise due to difference in designers' thinking and understanding [16]. It is claimed that transformation is a better way for construction of multi-dimensional schema (star schema) [15, pp-9], because it provides a step by step designing procedure. These steps enable a designer to design dimensional model, with-out having considerable knowledge of DM designing steps [5, pp-32]. In our previous experiments [14] (which are not a part of this paper) we found that, different individuals have different understandings of their ER model, so without using a transformation approach they may end up producing different dimensional models. Therefore, the transformation steps that enable designer to produce uniform Star Model (as output) from the same ER model are needed.

3. SHORTCOMINGS OF EXISTING TECHNIQUES

There are a number of approaches for transforming an ER schema into a Star Schema, however in this study, we have listed shortcomings of the manual approaches. The approaches are: *Schema Transformation approach* **(ST)** [6]**,** *E/R to Dimensional Fact Model* **(ERDFM)** [7]**,** *From ER to Dimensional model* **(GER)** [8]**.** Due to space limitations we briefly present shortcomings of each approach without any description of each approach.

Schema Transformation approach **(ST):** Shortcomings of this approach are: i) the process of random selection of transformation step (from T1… T14) is erratic, ii) the process is uncontrolled, as no guidelines are available for a designer, to select and apply a transformation step, iii) the approach is only applicable for fully normalized schema, iv) the approach proceeds without gathering user requirements, v) in this approach if fact attribute is not present in any entity, then it is not possible to identify aggregates of those facts.

E/R to Dimensional Fact Model **(ERDFM):** Shortcomings of this approach are: i) ERDFM is semi-automated approach and it produces best result only when schema is normalized, ii) in the absence of relationship between entity and selected fact, this technique cannot produce any attribute tree, iii) constellation of multiple start schemas' cannot be produced, by using this approach.

From ER to Dimensional model **(GER):** The shortcomings are: i) this approach produces surprisingly different Star models from same ER schema (that has differently designed ER Diagrams), ii) all transactional entities produce fact tables. In some cases the produced fact table doesn't make sense.

4. THE PROPOSED TRANSFORMATION APPROACH

In order to develop a Star model from ER model, a comprehensive method is required. Some researchers [6, 7, 8] have attempted to formulate ER to Star transformation technique, but these techniques have a number of deficiencies, identified in the previous section. To overcome these limitations we need a simplified, concrete and accurate transformation method that, i) takes user requirements, as starting point, ii) can be applied on any of the three types of ER models, identified in [14]. The types are, normalized, un-normalized or partially normalized ER models, iii) produces a single star schema, iv) takes minimum effort of designer, to translate ER to star model and v) can produce uniform star schema, vi) can produce star model equal to the star model produced by experienced DW designer. In the rest of this paper we use the name of our approach i.e. *ER to single star model*, abbreviated as ERSSM.

In the remaining part of this section, we briefly present our proposed transformation approach in the form of three algorithms. A few concepts in the algorithms are less described due to the following two reasons. i) These steps can be carried out in several different ways, ii) to avoid the discussion on advantages and disadvantages of each alternative and to keep focus on the major steps of the transformation approach.

Let β_i represents a business process that belongs to $\boldsymbol{\beta}$. Where, $\boldsymbol{\beta}$ is the set of all operational business processes that are represented in an entity relationship diagram ξ of an enterprise. $\{\beta_i$: i^{th} process that Є $\boldsymbol{\beta}$ derivable from IO(ξ) }

{ξ is a collection of entities (E_i - E_n, E_j & relationships) such that E_i has relationship r with E_j }

Also, *rel* Є *Rel* where is Rel is set of relationship types in ER [1:M, M:M, M:1]

For each business process β_i **,** perform the following steps.

Algorithm 1. Attributes categorization

Input: Reports for business process

Let R be the set of reports, such that each R_i can be generated from ξ using queries Q_i.

Q_i includes

i) Query definition, a combination of select, from, where, and, or, having etc.

ii) Schema metadata (M_i) used in the query. These are names of attributes, table names etc.

iii) Aggregate functions, used in the query. For example sum, max, min, count etc.

Create candidate-report-groups (CRG) of queries G_{qi},

where q, i represents group identification for queries, number of process.

Place the queries that belong to one business process in a group

For every report { If Query Є β_i business process, place it in CRG_i

//CRG_i is candidate report for identification of facts for i^{th} process

Else Query is miscellaneous }

For each query perform

{ Semantic analysis, by trimming of reports

Categorize attributes, i) slant, ii) critical attributes (derived) }

Algorithm 2. Transformation tree construction

Input: ER model & its critical attributes

```
            Marking of attributes
{   If        (attributes are in ER)              Mark attributes
  Else     {  Identify attributes from ER that contribute to generation of critical attribute using metadata $M_i$
    Mark identified attributes
$A_m$ represents marked attributes }
Entities classification
{            Create two entity groups $G_T$, $G_N$
//transactional & non-transact. entity group,
For each entity { Identify      relevant group
           Place      the query in its relevant group, ($G_T$ / $G_N$)    }
 Check:  If  (marked attributes belong to $G_T$ or relationship)
                     critical-attributes-identification is correct
            Else       Incorrect identification has taken place  }
            Build transformation tree
            {  Collect    marked attributes
            Formulate  fact-entity ($F_E$) with appropriate name
            Connect   critical attributes to $F_E$
For ER $\xi$ , {Find transactional entity with max no of relations & $A_m$
  Replace            transactional entity by $F_E$
  Select    $F_E$ as 'Stem' of transformation tree
  Removeremaining all member of $G_T$
Remove   tables which don't have any direct/in-direct relationship with $F_E$
   Attach  all the entities with $F_E$, they are 'Branches'
  Add       surrogate key $K_S$ to each entity connected to $F_E$
 For  each addition of $K_S$
 Add                  same attribute to $F_E$
Formulate primary key        by combining all $K_S$ added to $F_E$
 Swap      existing relationship of $F_E$, with newly born $K_S$     }
 Check:  If          (number of attribute is primary key $P_K$ is equal to $K_S$ added   AND       number of attributes in primary key is equal to number of relations)
correct transformation tree
Else        Incorrect transformation tree construction             }
            Identify    hierarchies for relationship
If          a hierarchies create circular loop $C_L$
             for each $C_L$
             {        Break      $C_L$ into two tables
Add         Surrogate key to newly born table $NB_T$
Add                   above attribute $A_N$ to $F_E$
 Amend    primary key by making $A_N$, as part of $P_K$ in $F_E$
 Remove   existing relationship of $NB_T$
 Create        new relationship between $F_E$ and $NB_T$ by using $A_N$}
Else        entities connected with $F_E$ form one dimension  }
For each dimension,
 { Collapse dimension           by de-normalizing each hierarchy
                Resolve              Naming conflicts }
Star schema $SS_i$ is generated for $i^{th}$ business process ($\beta_i$).
```

Following the preceding statement of algorithm 1, repeat algorithm 1&2 for all β_i's with the exception of miscellaneous group.

Algorithm 3. Constellation schema generation

Input: Star schema SS_i for each business process

```
Schema Constellation generation
{ Create  bus-metrics [1] between dimensions of $SS_i$, $SS_j$ ....$SS_n$ and identified facts
 Find  (relationships between $SS_i$ and $SS_j$ ) for all i & j belongs to input schemas
If           Relationships  are found
{ Combine              SS from i to j by using distinct dimensions
  Repeat the above statement for by adding single SS for each iteration   }
                                                        }
Else        Constellation cannot be generated        }
```

OUTPUT: Single constellation Star model.

5. EVALUATION: A CASE STUDY

In order to provide proof of the concept, we use an example study (see fig 1) to present applicability of the proposed technique. This case study's ER Diagram is fully normalized and it is modified form of the north wind database available in SQL Server 2000. We use this case study to completely show the versatile features of our approach. This section briefly presents step by step implementation of the proposed technique. The technique starts from collecting candidate reports and grouping them together. The details are as follows:

Candidate report groups: In the first step candidate reports are collected. Here, we have taken a number of reports and divided them into the following four groups i) sales, ii) purchase, iii) salaries and iv) miscellaneous.

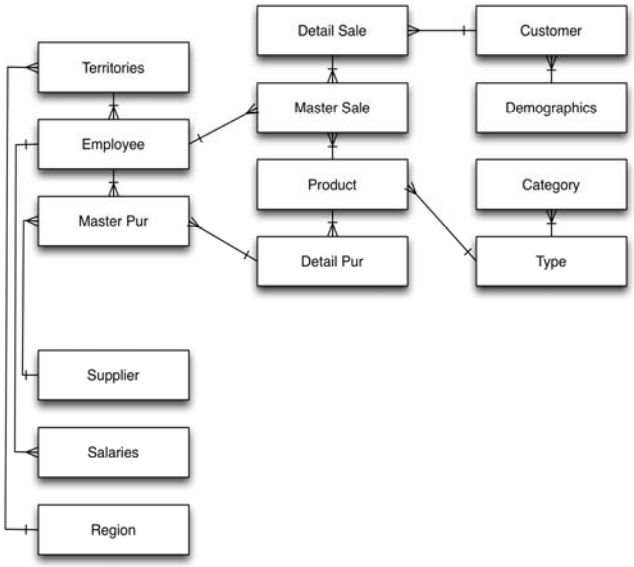


Figure 1. Sample ER Diagram

Semantic analysis: After semantic analysis and trimming of reports, critical attributes (to be called facts) are identified. Critical attributes belong to three categories, sales related, purchase related, salaries and miscellaneous. Sales related attributes are: Qty_sold, Amount_earned and total_sales_order. Similarly, purchase and salaries related attributes are collected. Slant attributes are: time, product_name, month, year, product_type, product category, customer name, location, address. Similarly, attributes for purchase and salaries are identified.

Marking of attributes: Some critical attributes are not available in the ER model, instead, they are produced by some operations on attribute(s) of ER model. In this phase the attributes that contribute to the production of critical attributes are marked.

In our example, Qty_sold attribute is not available in ER model, which is produced by sum(*Qty*). So mark qty attribute and similarly mark price attribute. Continue with purchase group and mark the following attributes in relation with purchase.

Entities classification: Entities are classified into two types G_T, G_N.

G_T = {Salaries, Sales_fact, Purchase_fact}, G_N = {Customer, employee, product, supplier, demographics, territory type, type category}

Transactional entities for sales are {customer, product}, for purchase {product, supplier} and for salaries {salaries}.

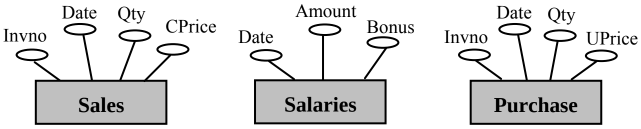


Figure 2. Fact Entities with Attributes

Build transformation tree:

Collect marked attributes and formulate a fact-entity. Three fact entities are formed F_{sales}, $F_{purchase}$, $F_{salaries}$, as shown in fig. 2.

Redraw ER model by inserting one fact entity at a time in place of the relationship entities (also called transactional entity). Also remove the entities that are not in relation with the newly inserted fact entity. In our example, we redraw ER model for fact entity *'Sales'*. Entities to be removed are crossed from the diagram. Repeat this process for *'purchase'* and *'salaries'* fact entities.

The transformation tree for sales shown in fig.3 has the following entities that form *stem*, Sales, employee, customer, product. Sales transformation tree will be of one shape what ever the form of ER model is. The *branches* are: i) demographics, ii) type, & category, iii) supplier, iv) territories and region. Similarly, transformation tree for purchase, and salaries are also drawn (but not shown in this paper).

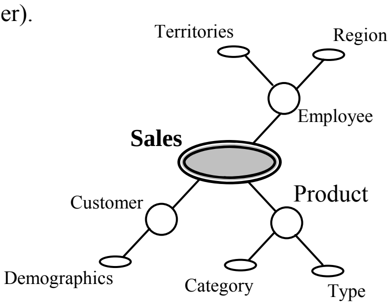
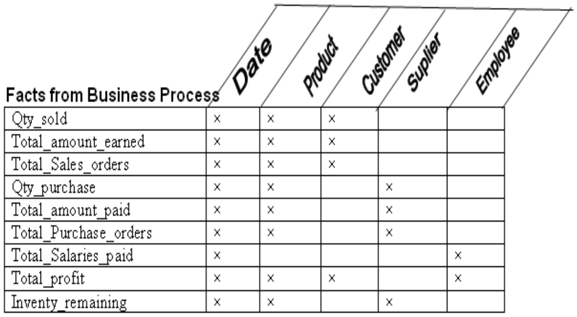


Figure 3. Transformation tree for Sales

Model the star schema by formulating primary and foreign keys. Add surrogate key to each dimension table and reformulate the relationship between fact and dimension table by using surrogate key. Crumple each dimension by collapsing hierarchies. Use de-normalization for this purpose and resolve conflicts. Dimensional model is finally produced. Repeat the same process for transformation tree for purchase and salaries. This way we have produced three star schemas, one for each transformation tree.

Constellation Schema generation: In order to produce a single constellation schema (constellation schema) from multiple star schemas we use Kimball's data warehouse bus architecture. Fig.4 shows the dimensions in relation with facts.



Facts from Business Process	Date	Product	Customer	Suplier	Employee
Qty_sold	×	×	×		
Total_amount_earned	×	×	×		
Total_Sales_orders	×	×	×		
Qty_purchase	×	×		×	
Total_amount_paid	×	×		×	
Total_Purchase_orders	×	×		×	
Total_Salaries_paid	×				×
Total_profit	×	×	×		×
Inventy_remaining	×	×		×	

Figure 4. A simple data warehouse bus matrix

Generated Constellation: The constellation schema generated by using the values from bus architecture is given in Fig.5.

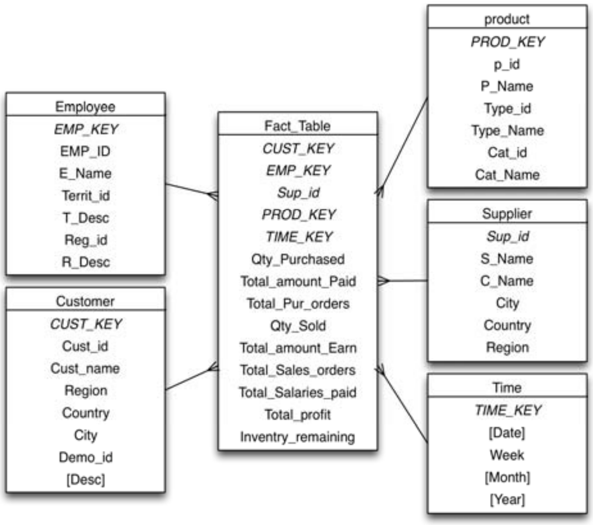


Figure 5. Single Star Schema

6. EXPERIMENTATION

For evaluation of the proposed technique experiments are performed whose details are presented in this section. Getting inspired from Corral's work [16], the outputs (star models) are compared to evaluate the differences. For comparison a set of parameters are used. The parameters used for comparison are: number of star schemas, fact tables, identified facts, dimensions tables, surrogate key, dimension attributes.

In this study, we intend to compare the star model, produced by experienced designer with the star model produced by inexperienced designers. Selection of the participants for the experiment is based on the following criteria: *a) experienced participant*, a participant with more 3 years of data warehouse designing experience in software industry or five years of university level teaching experience (in data warehousing) is considered as an experienced designer. *a) inexperienced participant*, a Masters in Computer Science student, who is participating in data warehousing course. These participants don't have any data warehousing teaching experience and they have never worked in software industry on data warehousing project/s. A total of 50 designers (25 experienced and 25 inexperienced DW designers) participated in this study.

A pool of 25 case studies, randomly collected from published papers, websites and books was used for the experimentation. The smallest case study had 11 entities whereas the largest case study had 25 entities. Some ER models used in the study were normalized, whereas the others were un-normalized.

Purpose: By using the proposed technique, to what extent inexperienced designers can produce star model similar to experienced designer. Lesser the difference better is the performance, of the proposed technique.

For data collection, textual explanation of a business case, its ER model and reports were provided to experienced designers and they were asked to develop a star model. In contrast, inexperienced designers were lectured, to completely explain our ERSSM transformation approach. The lecture was followed by a question-answer session to ensure that everyone understands the transformation approach correctly. ER models of the case studies (used by experienced designer for star modeling), were randomly distributed amongst inexperienced participants and designers and they were asked to develop star model. Hence, this experiment includes star models from 25 experienced and 25 inexperienced participants. The 50 star models, produced by participants, were used to get readings of the parameters to be used for comparison (number of star schemas, fact tables, identified facts, dimensions tables, surrogate key, dimension attributes).

7. ANALYSIS AND DISCUSSION

In order to analyze the variance between star models produced by experienced designer (using conventional approach, indicated by SCM) and star models produced by inexperienced designers (using our proposed approach, indicated by ERSSM), statistical methods for comparison of diagrams has been used i.e. ANOVA test for measuring variation between diagrams.

As discussed in section 6, seven parameters are used to compare star models. For this reason, variance is required to be calculated for each parameter. This motivates the need to apply ANOVA test separately for each variable. Acquired values are given in table 1. The values shown in table 1, clearly indicate that star models produced by using ERSSM and SCM are identical. Table 1 gives p-value while adjusting alpha value to 0.05.

Table 1. Results of ANOVA Test

	SS	MS	F	P-value
Star schemas produced	2	2	6.857	0.011779
Distinct fact tables	2	2	6.857	0.011779
Distinct facts	24.5	24.5	4.558	0.037895
Fact attributes	21.78	21.78	2.346	0.132158
Distinct dimension	0	0	0	1
Dimension attributes	100.82	100.82	1.361	0.2490398
Surrogate keys	23.12	23.12	15.67	0.000248

ANOVA test is also used to find out average values and variance between ERSSM and SCM for each parameter. On an average, 17.2 distinct dimension attributes are produced by ERSSM and 20.1 by SCM. Similarly, average of star schemas produced by ERSSM is 1 and by SCM are 1.4. ERSSM produces 4.6 facts where as SCM produced 3.2 facts. The average values and variance given in table 2 indicates that star models produced by ERSSM and SCM are identical.

Table 2. Comparison of ERSSM and SCM

		ERSSM	SCM
Star schemas produced	Average	1	1.4
	Variance	0	0.58
Distinct fact tables	Average	1	1.4
	Variance	0	0.58
Distinct facts	Average	4.6	3.2
	Variance	7.08	3.67
Distinct fact attributes	Average	8.48	7.16
	Variance	11.8	6.81
Distinct dimensions	Average	4	4
	Variance	1.5	1.08
Dimension attributes	Average	17.2	20.1
	Variance	58.9	89.2
Surrogate keys	Average	2.64	1.28
	Variance	1.74	1.21

Data collected from the experiment is further used to calculate percentage of similarities between the star models produced by experience and inexperienced designers. For instance, in order to calculate percentage of similarities between star models for the variable *'number of dimension tables'*, we have identified the number of dimensions common between the star models produced by experienced and inexperienced designers. Its percentage is then calculated from the total number of dimensions.

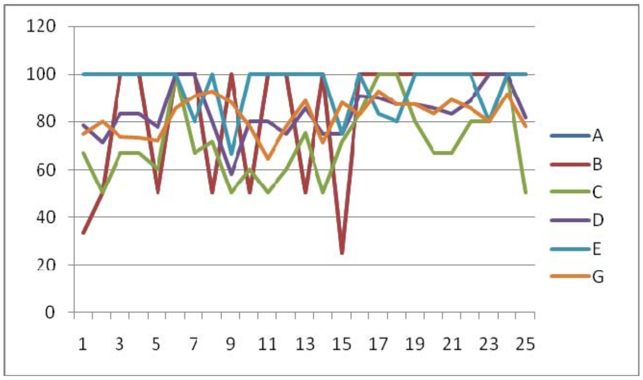


Figure 6. Similarities between the star models produced by ERSSM and SCM. In the graph number of star schemas, fact tables, identified facts, dimensions tables, surrogate key, dimension attributes are represented by A, B, C, D, E and G, respectively.

Similarly, we have repeated this process separately for all case studies and for each variable. The graph given in figure 6, shows the percentage of similarities for each variable. For example, B gives similarities in terms of *'number of distinct facts'*, between star models produced by experienced and inexperienced designers. In the graph values for the variables are separated by different colored lines. It is important to note, that most of the values vary from 60 to 100 percent. This variation of values clearly reflects that, star models produced from inexperienced designed (by using our proposed transformation approach) is similar to the star models produced by experienced designers.

On analyzing, it is found that, an average of 84.3% fact tables produced by experienced and inexperienced are identical. On an average, 70.8% facts are identical, 84% fact attributes are identical, 82.4% dimension attributes are identical and an average of 97.9% dimensions are identical. These average values clearly reflect similarities between the star model produced by experienced designers and inexperienced designers (who use our proposed approach). Only 42.2% distinct surrogate keys are identical. This is because of the reason that, we added surrogate key for each relationship between dimension and fact tables. The added surrogate key is based on the recommendation of Pasha et al [17]. It is said that surrogated key enhances flexibility in star model, supports business dynamics and evolutionary nature of a star model.

8. CONCLUSION

The paper surveys existing approaches used to transform ER model to Star model. Shortcomings of the existing approaches are identified and motivation for using transformation is given. A systematic approach is then suggested to transform ER model to Star model is presented with the help of three algorithms. An example ER model is used to demonstrate the applicability of the proposed approach.

In order to measure the quality of Star model produced by our transformation approach, we have conducted an experiment and compared the Star models produced by experienced data warehouse designer with the Star models produced by inexperienced designers (by using our transformation approach). By analyzing the data collected from the experiments it is proved that difference between the Star models is small. Hence, inexperienced designer, by using our transformation approach, can develop a Star model identical to the experienced designer.

The advantages of the proposed transformation approach are as follows: i) inexperienced designer can produce a data warehouse very similar to that of experienced designer, ii) produces equally good results for normalized and un-normalized ER models, iii) different star model from the same ER schema are not produced,

The three major limitations of this study are, i) some concepts in the algorithms are not fully explained in this paper, ii) the empirical study is based on a small sample 50 participants, iii) complex ER models are not used for the empirical study.

9. REFERENCES

[1] Ponniah, P., Data Warehousing Fundamantal, New Jersey, June 2001

[2] Ramakrishnan,R., Gehrke,J., DataBase Management Systems,3rd Edition, New York, 2003

[3] Connolly,T., Begg,G., Database Systems, 2nd edition, Library of congress cataloging in Publication Data, 2001.

[4] Fred R. McFadden and Jeffrey A. Hoffer, Modern Database Management, 4th Edition, by the Benjamin/ Cummings PublisherCompany, Inc. 1994.

[5] Kimball, R., and Ross, M., The Data Warehousing Toolkit, John Wiley & Sons, 2002.

[6] Marotta, A. and Ruggia, R., Data Warehousing Design: A Schema-transformation Approach, Proceedings of 12th SCCC, Atacama, Chile, 2002.

[7] Golfarlli, M., Maio, D. and Rizzi, s., Conceptual Design of Data Warehouses from E/R Schemes, In proceeding of 31st HICSS, Hawaii, USA, 1998.

[8] Moody, D.L. and Kortink, M. A. R., From Enterprise Models to Dimensional Models: A Methodology for Data Warehousing and Data Mart Design, In Proceeding. 2nd DMDW, Toronto, Canada, 2002

[9] husemann, B., Lechtenborger, J. and Vossen, G., Conceptual Data Warehouse Design, In proceeding of DMDW'2000 Stockholm, Sweden, June 5-6, 2000

[10] Chen,Y. T. and Hsu, P.Y., A grain preservation translation algorithm: From ER diagram to multidimensional model, in format, Sci. (2007), doi: 10.1016/j.ins.2007.03.017.

[11] Dov Dori , Roman Feldman, Arnon Sturm. Transforming an Operational System Model to a Data Warehouse Model: A Survey of Techniques, Proceeding of the International Conference on Software- Science, Technology & Engineering (SwSTE'05).

[12] Chaudhuri,S. Dayal,U. An Overview of Data Warehousing and OLAP Technology. ACM SIGMOD Record, 26(1): 65-74, March 1997.

[13] Matteo Golfarelli, Stefano Rizzi, Lunis Cella. Beyond Data Warehousing: What's Next in Business Intelligence? In the proceeding of DOLAP' 04, 2004 Washington, DC, USA.

[14] Sohail Abid, From ER to Single Star Model (ERSSM): A systematic translation approach. MSCS thesis, February 2008 CIIT, Lahore, Pakistan

[15] Ballard, C., Farrell, M, D., Gupta, A., Mazuela, C., Vohnik, S., Dimensional Modeling: In a Business Intelligence Environment, First edition IBM 2006.

[16] Karen Corral, David Schuff, Robert D St.Lousi, The impact of alternative diagrams on the accuracy of recall: A comparison of star-schema diagrams and entity-relationship diagrams, Decision Support System, Vol. 42(1), pp.450-468, 2006.

[17] M.A.Pasha, J.A.Nasir, M.K.Shahzad, Semi-star schema for managing data warehouse consistency, Proceedings of IEEE National Conference on Emerging Technology, pp. 63-66, Karachi, Pakistan.