

From ER Models to Star Schemas: Mapping From Operational Databases to Data Warehouse Designs

Daniel L. Moody
Department of Cybernetics
Czech Technical University
e-mail: dmoody@labe.felk.cvut.cz

School of Business Systems,
Monash University
e-mail: dmoody@infotech.monash.edu.au

Mark A.R. Kortink
Kortink and Associates
Melbourne, Australia
e-mail: mark@kortink.com

Abstract

Dimensional modelling is a conceptual modelling technique developed for designing data warehouse structures. It has become the predominant approach to designing data warehouses in practice and has proven to be highly successful in developing database structures that can be used directly by end users. This paper examines the nature of dimensional modelling and its relationship to traditional Entity Relationship (ER) modelling. It shows that a dimensional model is just a restricted form of an ER model, and that there is quite a straightforward mapping between the two. Understanding the relationship between the two types of models can help to bridge the gap between operational system (OLTP) design and data warehouse (OLAP) design. It also helps to resolve the difficult problem of matching supply (operational data sources) and demand (end user information needs) in data warehouse design. Finally, it results in a more complete dimensional design, which is less dependent on the designer's ability to choose the "right" dimensions. The paper also reports some preliminary results from empirical testing of the approach.

Keywords

Dimensional model, Entity Relationship (ER) model, star schema, data warehouse, OLAP

INTRODUCTION

Dimensional Modelling

Dimensional modelling is a conceptual modelling technique developed for designing data warehouses (Kimball, 1996; 1997; 2002). Its objectives are to create database structures that end users can easily understand and write queries against, and to optimise query performance. It has become the predominant approach to designing data warehouses in practice and has proven to be a major breakthrough in developing database structures that can be understood and used directly by end users. Dimensional modelling is based on a single, highly regular data structure called a *star schema*. A star schema consists of one central table called the *fact table* (which forms the centre of the star), surrounded by a number of *dimension tables* (which form the points of the star) (Figure 1):

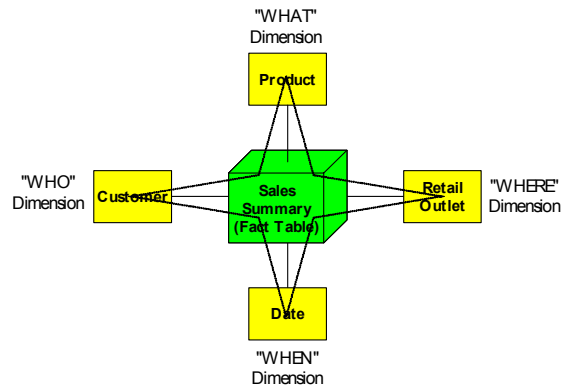


Figure 1. Star Schema Example

Fact tables generally correspond to particular types of business events: for example, orders, shipments, payments, bank transactions, airline reservations, hospital admissions. Such tables usually contain measures (e.g. dollar amounts, quantities), which may be analysed using numerical functions. Dimension tables provide the basis for aggregating data in the fact table. Dimensions typically answer “who”, “what”, “when”, “where”, “how” and “why” questions about the business events stored in the fact table.

Why Dimensional Modelling “Works” Heading Minor

Dimensional modelling is not based on any theory but has clearly been very successful in practice. The main reasons for its success are (Moody and Kortink, 2003b):

- It organises large amounts of data into cognitively manageable “chunks”. Each star schema consists of a small number of tables – typically less than “seven, plus or minus two” – which corresponds to the limits of human cognitive capacity (Miller, 1956; Baddeley, 1994). This helps to reduce complexity and consequent problems of information overload.
- Star schemas impose a hierarchical structure on enterprise data. This provides the ability to analyse data at different levels of detail, and to “roll up” and “drill down” in OLAP tools. Hierarchy is one of the most common ways of organising complexity for the purposes of human understanding (Flood and Carson, 1993; Simon, 1996; Klir and Elias, 2003).
- It simplifies formulation of queries by minimising the number of tables and therefore the number of joins required.
- It optimises query performance. A star schema has a fixed structure that has no alternative join paths, which greatly simplifies the evaluation and optimisation of queries (Raisinghani, 2000).

Objectives of this Paper

This paper examines the nature of dimensional modelling and its relationship to traditional Entity Relationship (ER) modelling. It shows that an ER model can be transformed into a set of dimensional models by a process of selective subsetting, denormalisation and (optional) summarisation. Understanding the relationship between the two types of models can help to bridge the gap between operational system (OLTP) design and data warehouse (OLAP) design.

PREVIOUS RESEARCH

Most of the previous research in dimensional modelling has focused on developing new conceptual modelling notations to represent multi-dimensional data. For example, Cabibbo and Torlone (1998) define a formal mathematical model called the *Multidimensional (MD) Model* to represent multidimensional data. Lehner (1998) proposes the *Nested Multidimensional Data Model (NMDM)*, which represents data at two different (“nested”) levels. Sapia et al (1998) define an extension to the ER model called the *Multidimensional E/R Model (ME/R)*. Golfarelli et al (1998, 1999) define a graphical conceptual modelling technique called the *Dimensional Fact Model (DFM)*, which is used to develop a complete data warehouse design methodology. Tryfona et al (1999) propose the *StarER Model*, which is an extension of the ER model to model multidimensional data. Sanchez et al (1999) propose a conceptual model called IDEA to model multidimensional data, which is used as the basis for a CASE tool (IDEA-DWCASE) and a data warehouse design methodology (EINSTEIN). Trujillo et al (1999, 2000) define an OO conceptual model called the *Object Oriented Multidimensional Model (OOMD)* based on a subset of UML. Franconi and Sattler (2000) define an

extension to the ER model called the *Data Warehouse Conceptual Model (CDWDM)* based on description logics. Husemann et al (2000) define a graphical technique for multidimensional modelling based on functional dependencies. Schelp (2000) defines a multidimensional conceptual model called *Kubenstrukturmodell (Cube Structure Model)*. Nguyen et al (2000) define an object-oriented multidimensional data model and use UML to define the model for the purpose of representing multidimensional data using OO databases. Lujan-Mora et al (2000) define an extension to UML to represent multidimensional data using the package construct. Finally (and rather appropriately!), Aballo et al (2000) propose a multidimensional conceptual model called *Yet Another Multidimensional Model (YAM)* based on UML.

At a high level, most of the models proposed are quite similar, in that they define constructs to represent facts, measures, dimensions etc. However none have become widely accepted in practice, and most of them do not appear to have been applied outside a research environment. There are also major practical and theoretical problems with the techniques proposed:

- Complexity management: Most of the models proposed represent data at the level of individual attributes – that is, each node in the graphical notations corresponds to a single attribute. This leads to an complexity explosion when dealing with real world data warehouses, where a single dimension may contain more a hundred attributes (Kimball, 2002). Such techniques seem to re-introduce the problems of complexity that dimensional modelling was designed to solve. Empirical studies show that people have difficulty understanding conceptual models when they exceed the limits of human cognitive capacity (seven, plus or minus two entities) (Moody, 2002).
- Ontological soundness: Wand and Weber (Wand and Weber, 1990; 1995; Weber, 1997) have proposed a theory of representation (referred to as the Bunge-Wand-Weber or BWW ontology) which defines a comprehensive set of ontological concepts needed to represent the real world. This provides a theoretical basis for evaluating and comparing different modelling notations (Green and Rosemann, 2000; Opdahl and Henderson-Sellers, 2002). One of the fundamental requirements of the BWW ontology is the need to distinguish between “things” and their properties – failure to do so has been found to reduce understanding of models (Shanks et al, 2003). Most of the multidimensional modelling methods proposed in the literature do not distinguish between entities and attributes, and are therefore *ontologically incomplete*.

Finally, we question whether such models are really necessary. While it might be interesting for researchers to develop new conceptual modelling techniques, it is important to establish the case for why they are needed. In this paper, we argue that new conceptual modelling techniques for dimensional data are unnecessary as a dimensional model is just a restricted form of ER model. Also, end users appear to be able to understand star schemas quite easily: it is therefore difficult to justify the need for a conceptual modelling technique as a “front end” to designing a star schema.

ER MODELS VS DIMENSIONAL MODELS

A New Paradigm?

There is a common misconception that dimensional modelling is fundamentally different to, and incompatible with, ER modelling. This is a view which has been energetically promoted by Ralph Kimball, who is widely regarded as the originator of dimensional modelling (e.g. Kimball, 1995; Kimball, 1996; 1997; 2002). As he says in his original book on data warehouse design (Kimball, 1996):

“Entity relation models are a disaster for querying because they cannot be understood by users and cannot be navigated usefully by DBMS software. Entity relation models cannot be used as the basis for enterprise data warehouses”

As a result, the starting point for most data warehouse design approaches is that you must forget everything you ever learnt about traditional database design. It is easy to understand Kimball’s motivation in promoting this view – the IT industry is driven by fads and fashions, and commercial success is driven by the ability to show that what one is promoting is “new”. However it is more difficult to understand why the academic community has simply accepted this premise at face value, as evidenced by their willingness to create new conceptual modelling techniques and even to define new normal forms (e.g. Lehner et al, 1998; Lechtenborger and Vossen, 2003; Levene and Loizou, 2003) for dimensional modelling.

We believe it is important to dispel this notion, as it holds back the development of the data warehousing field in two ways:

- It acts as a barrier for people trained in traditional database design techniques (ER modelling and normalisation) to learn data warehouse design. These techniques have been used to design database schemas for over two decades, and are a standard component of almost every university IT

curriculum(Thalheim, 2000). Clearly, it would be better to build on this existing knowledge base than to simply discard it – people learn better by building on what they already know.

- It acts as a barrier to building a “cumulative tradition” in the field(Kuhn, 1970; Keen, 1980; Weber, 1997). Rather than trying to establish a totally separate design discipline, data warehouse design should build on and link to existing body of knowledge in the database design field.

Dimensional Models as ER Models

As discussed earlier, most of the previous research in this area has focused on developing new conceptual modelling notations for dimensional data. However we argue that new conceptual modelling techniques are unnecessary, as a dimensional model is just a restricted form of ER model(Figure 2):

- There is a single entity called the fact table, which is in at least third normal form (3NF): violations to second normal form (2NF) would result in “double counting” in queries. The fact table forms an n-ary intersection entity (where n is the number of dimensions) between the dimension tables, and includes the keys of all dimension tables.
- There are two or more entities called dimension tables, each of which is related to the fact table via one or more one-to-many relationships. Dimension tables have simple keys, and are in at least 2NF: transitive dependencies (3NF violations) are allowed, but partial dependencies (2NF violations) and repeating groups (1NF violations) are not.

As we will show in the next section, there is quite a straightforward transformation between a normalised ER model and a dimensional model.

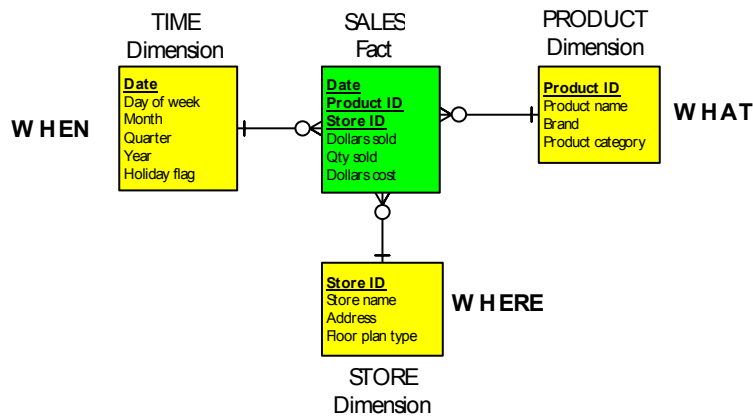


Figure 2. ER Representation of a Star Schema

FROM ER MODELS TO DIMENSIONAL MODELS

In this section, we describe how to derive dimensional models from normalised ER models. This provides a way of repackaging data stored in operational systems into a form that end users can easily understand and write queries against. Deriving dimensional models from an ER model provides a more structured approach than starting from first principles, which can help to avoid many of the pitfalls faced by inexperienced designers.

“Dimensionalising “ an ER Model

The transformation of an ER Model to a set of dimensional models takes place in four steps:

1. Classify entities
2. High level star schema design
3. Detailed fact table design
4. Detailed dimension table design

Step 1 Classify Entities

The first step in the transformation procedure is to classify entities into three distinct categories:

- **Transaction Entities:** These are entities which record details of business events e.g. orders, shipments, payments, insurance claims, bank transactions, hotel bookings, airline reservations and hospital

admissions. It is such events that most decision support applications are centred around, in order to identify patterns, trends, opportunities and potential problems in business operations.

- **Component Entities:** These are entities which are related to a transaction entity by a one-to-many relationship. These are entities which are directly involved in the business event, and define details of “who”, “what”, “where”, “how” and “why”.
- **Classification Entities:** These are entities which are related to a component entity by a chain of one-to-many relationships. These define embedded hierarchies in the data model.

Example

To illustrate the approach, we use an example ER model for a conference/event organiser (Figure 3). This consists of 34 entities, which is only about a third of the size of the average operational data model, but large enough to illustrate the main principles of the approach. Even in a model of this size, the problems of complexity are clearly evident. Most end users would find such a schema incomprehensible as it exceeds human cognitive capacity many times over. Even quite simple queries will require multi-table joins, which are beyond the capability of most end users.

The classification of entities is shown in Figure 3.

- Transaction entities: these are usually relatively easy to identify. In the example, there are two: Event Registration (on the demand/revenue side) and Speaker Engagement (on the supply/cost side).
- Component entities: Event Registration has five components (Delegate, Booking Method, Discount Type, Promotion and Event) while Speaker Engagement has four components (Speaker, Event, Payment Level, Session Type). One of the component entities (Event) is shared by both transactions.
- Classification entities: There are 17 classification entities in the model, which define separate but partially overlapping hierarchies in the model.

Note that some entities do not fit into any of these categories. Such entities do not fit the hierarchical structure of a star schema, and therefore cannot be represented in dimensional form. The process of “dimensionalising” an ER model effectively “weeds out” non-hierarchical data.

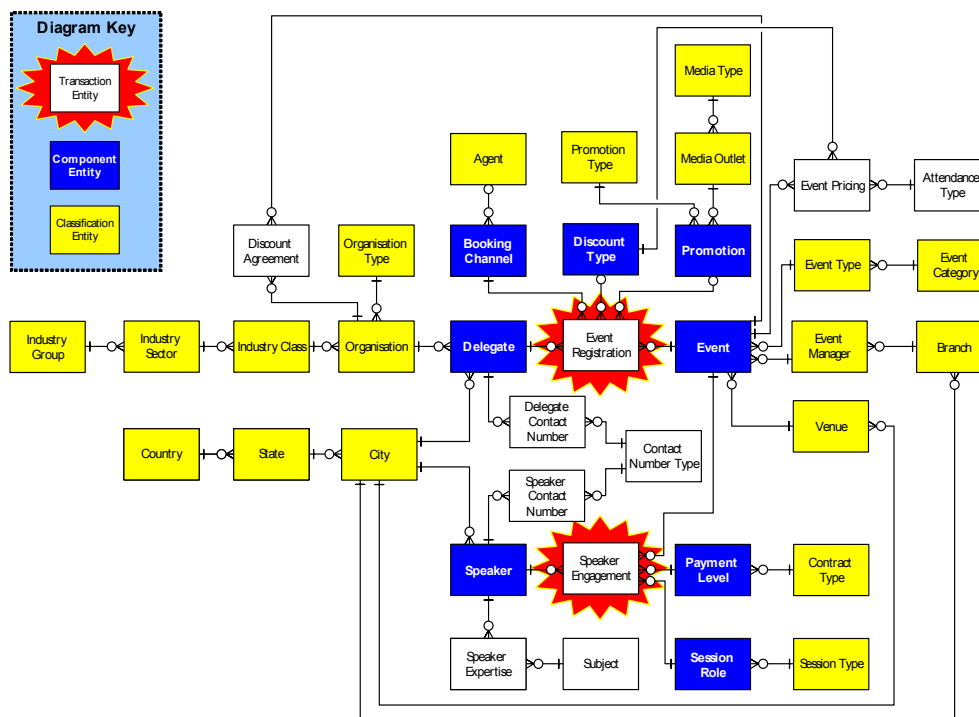


Figure 3. Classification of Entities

Step 2: High Level Star Schema Design

In this step, relevant star schemas are identified and their high level structure defined (entity level design).

2.1 Identify Star Schemas Required

Each transaction entity is a candidate for one or more star schemas. Each star schema should be centred around a single business event, so that it represents a manageable sized “chunk” of data. However there is not a one-to-one correspondence between transaction entities and star schemas:

- Not all transactions may be important for decision support purposes: user input will be required to choose which transactions are relevant.
- Multiple star schemas at different levels of detail may be required for a particular transaction.

2.2 Define Level of Summarisation

One of the most critical decisions in star schema design is to choose the appropriate level of *granularity*, which is the level of detail or summarisation at which data is stored (Inmon, 1996). At the highest level, there are two main options in choosing the level of granularity:

- Unsummarised (transaction level granularity): this is the highest level of granularity, where each fact table row corresponds to a single transaction or line item.
- Summarised: transactions may be summarised by a particular subset of dimensions or dimensional attributes. In this case, each row in the fact table corresponds to multiple transactions.

The lower the level of granularity (the higher the level of summarisation), the less storage space required and the faster queries will be executed. However the downside is that summarisation loses information and therefore limits the types of analyses that can be carried out. Transaction level granularity provides maximum flexibility for analysis, as no information is lost from the original normalised model. However for performance reasons or to simplify the view of data for a particular group of decision makers, some level of summarisation may be required. In practice, multiple star schemas at different levels of granularity are generally created for each transaction entity, to suit the needs of different users.

2.3 Identify Relevant Dimensions

The component entities associated with each transaction entity represent *candidate dimensions* for the star schema. However there is not a one-to-one correspondence between component entities and dimensions:

- Not all components may be relevant for purposes of analysis, or for the level of granularity chosen.
- If a component entity has no dependent attributes, its key will be included in the fact table but it will not be represented by an explicit dimension table. This is called a *degenerate dimension*.
- Explicit dimensions are required to represent Date and/or Time, to support different levels of historical analysis. Date and Time are not normally explicitly represented in operational systems, as they are handled by built-in DBMS functions. These correspond to *data types* rather than entities at the operational level.

Figure 4 shows the transaction level star schema for the Event Registration entity. Each row in the fact table corresponds to an individual event registration (transaction level granularity). The star schema has five dimension tables, corresponding to four of the five component entities plus a time dimension (Date). There are multiple relationships to the date dimension corresponding to the different dates recorded for each registration. One of the component entities (Discount Type) is a degenerate dimension as it contains no dependent attributes.

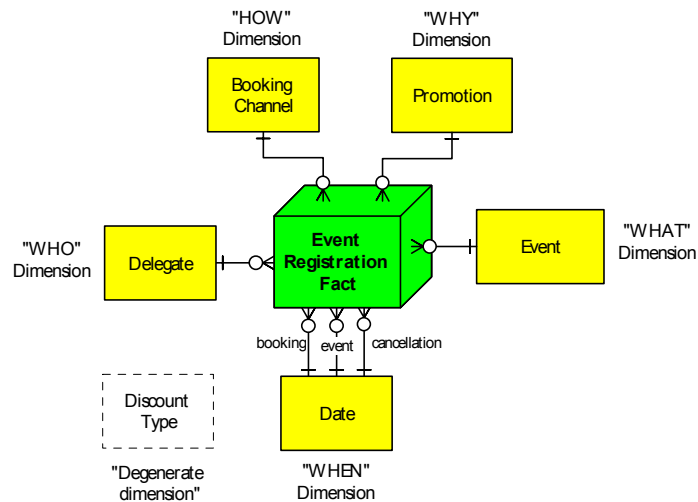


Figure 4. Transaction Level Star Schema for Event Registration Transaction

When non-transaction level granularity is chosen, the dimensions required will be determined by how the transactions are summarised. This will generally be some subset of the dimensions used in the transaction level star schema, although dimensions may be subsets of the transaction level dimension tables.

Step 3: Detailed Fact Table Design

In this step, we complete the detailed (attribute level) design for fact tables in each star schema.

3.1 Define Key

The key of each fact table is a composite key, consisting of the keys of all dimension tables plus any degenerate dimensions.

3.2 Define Facts

The non-key attributes of each fact table are measures (facts) that can be analysed using numerical functions. These are derived from attributes in transaction entities. A key concept in defining facts is that of *additivity* (Kimball, 1996; 2002):

- *Fully additive facts*: these are facts that can be meaningfully added across all dimensions. In the Event Registration example, Registration Amount (\$) can be added across any combination of dimensions to get total sales for a particular day, event, delegate or booking channel.
- *Semi-additive facts*: these are facts that can be meaningfully added across some dimensions but not others (usually time).

Non-additive facts: these are facts that cannot be meaningfully added across any dimensions. In the Event Registration example, Discount Percentage (%) cannot be added across any dimensions.

Wherever possible, additive facts should be used to prevent errors in queries. This means converting semi- and non-additive facts to additive facts. In the example, Discount Percentage (%) (a non-additive fact) can be converted to an additive fact by multiplying it by Registration Amount (\$) to form Discount Amount (\$). Note that this transformation does not lose information: Discount Percentage can be derived by dividing Discount Amount price by Registration Amount.

Figure 5 shows the transaction level fact table for the Event Registration transaction entity. The key of the table is a composite key, consisting of the keys of all dimension tables plus the degenerate dimension. Each row in the fact table corresponds to an individual event registration and contains all attributes in the original transaction entity. In this case, no data is lost from the original (normalised) data model. However two non-additive facts (Discount Percentage and Refund Percentage) have been converted to additive facts.

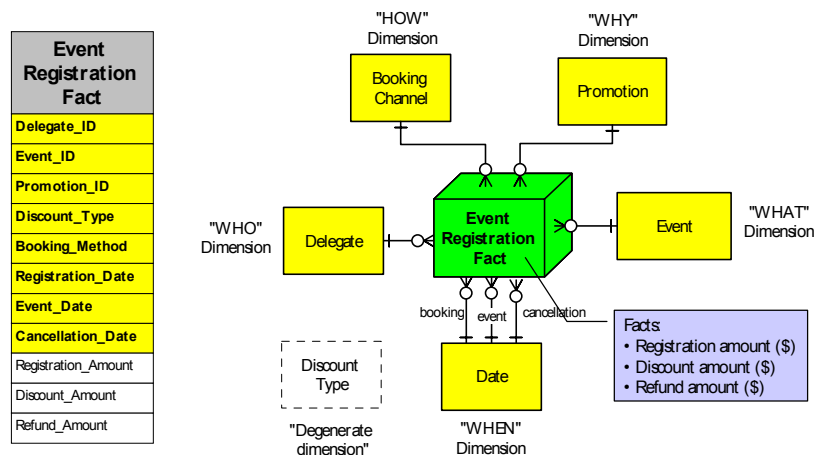


Figure 5. Event Registration Fact Table

Step 4: Detailed Dimension Table Design

In this step, we complete the detailed (attribute level) design for dimension tables in each star schema.

4.1 Define Dimensional Key

The key of each dimension table should be a simple (single attribute) key. In most cases, this is just the key of the underlying component entity. However sometimes the operational key may need to be generalised, to ensure that it remains unique over time. Operational systems often only require uniqueness at a point in time, which may cause problems in the data warehouse environment when performing historical analysis. Another situation where the key may need to be generalised is the case of *slowly changing dimensions*, in which different “states” of the component entity are recorded over time (Kimball, 1996; 2002).

4.2 Collapse Hierarchies

Dimension tables are formed by denormalising hierarchies (defined by classification entities) into component entities. The resulting dimension table consists of the union of all attributes in the original entities – as many descriptive attributes as possible should be included to support analysis of data in different ways. This process introduces redundancy in the form of *transitive dependencies*, which are violations to third normal form (3NF) (Codd, 1970). This means that the resulting dimension table is in second normal form (2NF).

4.3 Augment or Replace Codes and Abbreviations by Descriptive Text

To make the star schema as understandable as possible, codes and abbreviations in the source data should be augmented or replaced by descriptive text (Kimball, 1996; 2002). While codes and abbreviations play an important place in efficient processing of transactions at the operational level, they generally obfuscate things in the data warehouse environment – they place additional cognitive load on the end user to remember what they mean.

CONCLUSION

Summary

This paper has described an approach for deriving dimensional models from ER models. This provides a “bridge” between operational system (OLTP) design and data warehouse (OLAP) design. It also helps to resolve the difficult problem of matching “supply” (operational data sources) and “demand” (end user information needs) in data warehouse design. Finally, it results in a more complete dimensional design, which is less dependent on the designer’s ability to choose the “right” dimensions.

We have challenged the widely accepted view that dimensional modelling is fundamentally different to and incompatible with ER modelling. We have shown that a dimensional model is just a restricted form of ER model and there is quite a straightforward transformation between the two. An ER model can be transformed into a set of dimensional models by a process of selective subsetting, denormalisation and (optional) summarisation:

- Subsetting: The data contained in the ER model is partitioned into a set of separate star schemas, each centred around a single business event (transaction entity). This reduces complexity through “chunking”.

- Denormalisation: Hierarchies in the ER model are collapsed to form dimension tables. This further reduces complexity by reducing the number of tables.
- Summarisation: The most flexible dimensional structure is one where each fact represents a single transaction or line item. However summarisation may be required for performance reasons, or to suit the needs of a particular group of users.

Validation of the Approach

It is essential for IS design methods to be validated in practice – ultimately, the scientific merit of any method is an *empirical* rather than a theoretical question (Rescher, 1973; Ivori, 1986). However IS design research tends to emphasise the development of new methods while addressing the use and evaluation of methods in only a limited fashion (Bubenko, 1986; Curtis, 1986; Fitzgerald, 1991; Westrup, 1993; Wynkoop and Russo, 1997; Moody and Shanks, 1998). The method described in this paper has been empirically tested in three ways.

Field Testing: Action Research

Action research provides a method for testing and refining research ideas by applying them in practice (McCutcheon and Jurg, 1990; Jönsson, 1991; Baskerville and Wood-Harper, 1996; Hatten et al, 1997). Using this approach, the method has been applied over a three year period in data warehousing projects in a range of industries including health, banking, insurance, manufacturing and utilities. These projects have provided real world tests of the effectiveness of the method. The method was originally defined in (Moody and Kortink, 2000) but has been refined considerably since this time. Most of the refinements have been to deal with exceptions and special cases that arise in practice (e.g. non-hierarchical data, slowly changing dimensions, minidimensions, heterogeneous star schemas). For reasons of space, the method is only presented in simplified form here but is described in detail in (Moody and Kortink, 2003b; 2003a).

Field Testing: Teaching to Practitioners

The method has also been taught to practitioners via public courses. This is an important step in the evolution of any method: from a method that the authors can use to one that anyone can use. Feedback from teaching experiences has been used to improve the method, particularly in making the details of the transformation approach more explicit. Teaching practitioners is more likely to result in useful feedback than teaching university students, as students lack the knowledge and experience to evaluate the practical applicability of a method and identify potential weaknesses.

Laboratory Testing: Practitioner Acceptance Testing

Regardless of the potential benefits of design methods, unless they are used in practice, these benefits cannot be realised. Adoption in practice is an important pragmatic measure of method “success” and of the impact of research on practice (Fitzgerald, 1991; Moody, 2003). A laboratory experiment was conducted in which experienced practitioners were trained to use the method, applied it to a range of problems and asked to provide their perceptions of it via a post-task survey. This provides a form of “practitioner acceptance testing” of the method, analogous to user acceptance testing which is routinely conducted for information systems. The study showed significantly positive results for perceived ease of use, perceived usefulness and intention to use, which suggests that the method has a high likelihood of being adopted in practice.

REFERENCES

- ABALLO, A., SAMOS, J. and SALTOR, F. (2002): “YAM (Yet Another Multidimensional Model): An Extension of UML”, Proceedings of the International Database Engineering and Application Symposium,
- BADDELEY, A.D. (1994): “The Magical Number Seven: Still Magic After All These Years?”, *Psychological Review*, 101, 2.
- BASKERVILLE, R.L. and WOOD-HARPER, T. (1996): “A Critical Perspective on Action Research as a Method for Information Systems Research”, *Journal of Information Technology*, 3, 11, pp. 235-246.
- BUBENKO, J.A. (1986): “Information Systems Methodologies - A Research View”. *Information Systems Design Methodologies: Improving The Practice*, T.W. Olle, H.G. Sol and A.A. Verrijn-Stuart, (Eds.), North-Holland.
- CABIBBO, L. and TORLONE, R. (1998): “A Logical Approach to Multidimensional Databases”, Proceedings of the 6th International Conference on Extending Database Technology (EDBT'98), H.-J. Schek, F. Saltor, I. Ramos and G. Alonso, (Eds.), Valencia, Spain, Springer, March 23-27.

- CODD, E.F. (1970): "A Relational Database Model For Large Shared Data Banks", *Communications of the ACM*, 13, 6, pp. 377-387.
- CURTIS, B. (1986): "By The Way, Did Anyone Study Any Real Programmers?". *Empirical Studies of Programmers*, E. Soloway and S. Iyengar, (Eds.), Norward, N.J., Ablex.
- FITZGERALD, G. (1991): "Validating New Information Systems Techniques: A Retrospective Analysis", *Information Systems Research: Contemporary Approaches And Emergent Traditions*, H.E. Nissen, H.K. Klein and R. Hirschheim, (Eds.), North-Holland,
- FLOOD, R.L. and CARSON, E.R. (1993): *Dealing With Complexity: An Introduction To The Theory And Application Of Systems Science*, Plenum Press.
- FRANCONI, E. and SATTLER, U. (2000): "A Data Warehouse Conceptual Model for Multidimensional Aggregation", *Proceedings of the International Workshop on the Design and Management of Data Warehouses*, Stockholm, Sweden,
- GOLFARELLI, M., MAIO, D. and RIZZI, S. (1998): "Conceptual Design of Data Warehouses from E/R Schemes", *Proceedings of the Hawaii International Conference on Systems Sciences*, Kona, Hawaii, January 6-9.
- GOLFARELLI, M. and RIZZI, S. (1999): "Designing the Data Warehouse: Key Steps and Crucial Issues", *Journal of Computer Science and Information Management*, 2, 3, January 6-9.
- GREEN, P. and ROSEMAN, M. (2000): "Integrated Process Modelling: An Ontological Evaluation", *Information Systems Journal*, 25, 2.
- HATTEN, R., KNAPP, D. and SALONGA, R. (1997): "Action Research: Comparison with the Concepts of The Reflective Practitioner and Quality Assurance", *Action Research Electronic Reader (on-line)*, March.
- HUSEMANN, B., LECHTENBORGER, J. and VOSSEN, G. (2000): "Conceptual Data Warehouse Design", *Proceedings of the International Workshop on Design and Management of Data Warehouses*, Stockholm, Sweden, June.
- INMON, W.H. (1996): *Building the Data Warehouse (2nd Edition)*, J. Wiley & Sons, New York.
- IVARI, J. (1986): "Dimensions Of Information Systems Design: A Framework For A Long Range Research Program", *Information Systems Journal*, June.
- JÖNSSON, S. (1991): "Action Research", *Information Systems Research: Contemporary Approaches And Emergent Traditions*, H.E. Nissen, H.K. Klein and R. Hirschheim, (Eds.), North-Holland,
- KEEN, P.G.W. (1980): "MIS Research: Reference Disciplines and a Cumulative Tradition", *1st International Conference on Information Systems*, December.
- KIMBALL, R. (1995): "Is ER Modeling Hazardous to DSS?", *DBMS Magazine*, October.
- KIMBALL, R. (1996): *The Data Warehouse Toolkit*, John Wiley and Sons, New York.
- KIMBALL, R. (1997): "A Dimensional Manifesto", *DBMS Online*, August.
- KIMBALL, R. (2002): *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modelling*, John Wiley and Sons, New York.
- KLIR, G.J. and ELIAS, D. (2003): *Architecture of Systems Problem Solving (2nd Edition)*, Plenum Publishing Corporation, New York.
- KUHN, T.S. (1970): *The Structure Of Scientific Revolutions*, University Of Chicago Press.
- LECHTENBORGER, J. and VOSSEN, G. (2003): "Multidimensional Normal Forms for Data Warehouse Design", *Information Systems*, 28, pp. 415-434.
- LEHNER, W. (1998): "Modelling Large Scale OLAP Scenarios", *Advances in Database Technology: Proceedings of the 6th International Conference on Extending Database Technology (EDBT'98)*, H.-J. Schek, F. Saltor, I. Ramos and G. Alonso, (Eds.), Valencia, Spain, Springer, March 23-27.
- LEHNER, W., ALBRECHT, J. and WEDEKIND, H. (1998): "Normal Forms for Multidimensional Databases", *Proceedings of the 10th International Conference on Scientific and Statistical Data Management (SSDBM'98)*, Capri, Italy, July 1-3.
- LEVENE, M. and LOIZOU, G. (2003): "Why is the snowflake schema a good data warehouse design?", *Information Systems Journal*, 28, 5, pp. 225-240.

- LUJÁN-MORA, S., TRUJILLO, J. and SONG, I.-Y. (2002): "Multidimensional Modeling with UML Package Diagrams", International Conference on Conceptual Modelling (ER'2002), S. Spaccapietra, S.T. March and Y. Kambayashi, (Eds.), Tampere, Finland, October.
- MCCUTCHEON, G. and JURG, B. (1990): "Alternative Perspectives on Action Research", Theory into Practice, 24, 3, Summer.
- MILLER, G.A. (1956): "The Magical Number Seven, Plus Or Minus Two: Some Limits On Our Capacity For Processing Information", The Psychological Review, March.
- MOODY, D.L. (2002): "Complexity Effects On End User Understanding Of Data Models: An Experimental Comparison Of Large Data Model Representation Methods", Proceedings of the Tenth European Conference on Information Systems (ECIS'2002), Gdansk, Poland, June 6-8.
- MOODY, D.L. (2003): "The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods", Proceedings of the Eleventh European Conference on Information Systems (ECIS'2003), Naples, Italy, June 18-21.
- MOODY, D.L. and KORTINK, M.A.R. (2000): "From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design", Proceedings of the 2nd International Conference on Design and Management of Data Warehouses (DMDW'00), Stockholm, June 10.
- MOODY, D.L. and KORTINK, M.A.R. (2003a): "From ER Models to Dimensional Models: Bridging the Gap between OLTP and OLAP Design", Journal of Business Intelligence, 8, 3, Summer.
- MOODY, D.L. and KORTINK, M.A.R. (2003b): "From ER Models to Dimensional Models Part II: Advanced Design Issues", Journal of Business Intelligence, 8, 4, Fall.
- MOODY, D.L. and SHANKS, G.G. (1998): "Evaluating and Improving the Quality of Entity Relationship Models: An Action Research Programme", Australian Computer Journal, November.
- NGUYEN, T.B., TJOA, A.M. and WAGNER, R.R. (2000): "An Object Oriented Multidimensional Model for OLAP", Proceedings of the First International Conference on Web-Age Information Management (WAIM'2000), H. Lu and A. Zhou, (Eds.), Shanghai, China, Springer Lecture Notes in Computer Science 1846, June 21-23.
- OPDAHL, A.L. and HENDERSON-SELLERS, B. (2002): "Ontological Evaluation of the UML Using the Bunge-Wand-Weber Model", Software and Systems Modelling, 1, 1, pp. 43-67.
- RAISINGHANI, M.S. (2000): "Adapting Data Modelling Techniques for Data Warehouse Design", Journal of Computer Information Systems, 40, 3, Spring.
- RESCHER, N. (1973): The Primacy of Practice: Essays Towards A Pragmatically Kantian Theory Of Empirical Knowledge, Basil Blackwell, Oxford.
- SANCHEZ, A., CAVERO, J.M., DE MIGUEL, A. and MARTINEZ, P. (1999): "IDEA: A Conceptual Multidimensional Data Model and Some Methodological Implications", Proceedings of the Sixth Congreso Internacionale de Investigacion en Ciencias Computacionales, Cancun, Mexico,
- SAPIA, C., MARKUS, B., HOFLING, G. and DINTER, B. (1998): "Extending the ER Model for Multidimensional Paradigms", Proceedings of the 17th International Conference on Conceptual Modelling, Singapore, Springer, November.
- SCHELP, J. (2000): Konzeptionelle Modellierung mehrdimensionaler Datenstrukturen analyseorientierter Informationssysteme, Wiesbaden: Deutscher Universitäts-Verlag.
- SHANKS, G.G., MOODY, D.L., NUREDINI, J., TOBIN, D. and WEBER, R.A. (2003): "Representing Things And Properties In Conceptual Modelling: An Empirical Evaluation", European Conference on Information Systems (ECIS 2003), Napoli, Italy, June 19-21.
- SIMON, H.A. (1996): Sciences Of The Artificial (3rd edition), MIT Press.
- THALHEIM, B. (2000): Entity Relationship Modeling: Foundations of Database Technology, Springer, Berlin ; New York.
- TRUJILLO, J. and PALOMAR, M. (1998): "An Object-Oriented Approach to Multidimensional Database Conceptual Modelling", Proceedings of the First ACM International Workshop on Data Warehousing and OLAP (DOLAP), Washington, DC, USA,

- TRUJILLO, J., PALOMAR, M. and GOMEZ, J. (2000): "Applying Object-Oriented Conceptual Modelling Techniques to the Design of Multidimensional Databases and OLAP Applications", Proceedings of the First International Conference on Web-Age Information Management (WAIM'2000), H. Lu and A. Zhou, (Eds.), Shanghai, China, Springer Lecture Notes in Computer Science 1846, June 21-23.
- TRYFONA, N., BUSBORG, F. and BORCH, J. (1999): "StarER: A Conceptual Model for Data Warehouse Design", Proceedings of the 2nd ACM International Workshop on Data Warehousing and OLAP,
- WAND, Y. and WEBER, R.A. (1990): "An Ontological Model of an Information System", IEEE Transactions on Software Engineering, pp. 1282-1292, November.
- WAND, Y. and WEBER, R.A. (1995): "On the Deep Structure of Information Systems", Information Systems Journal, 5, pp. 203-223.
- WEBER, R.A. (1997): Ontological Foundations Of Information Systems, Coopers And Lybrand Accounting Research Methodology Monograph No. 4, Coopers And Lybrand, Melbourne, Australia.
- WESTRUP, C. (1993): "Information Systems Methodologies in Use", Journal of Information Technology, 8.
- WYNEKOOP, J.L. and RUSSO, N.L. (1997): "Studying Systems Development Methodologies: An Examination Of Research Methods", Information Systems Journal, 7, 1, January.

COPYRIGHT

Daniel L. Moody and Mark A.R. Kortink © 2003. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.