IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP5120

Dimensional Modeling: Identifying Patterns and Classifying Patterns

Mary Elizabeth "M.E." Jones and Il-Yeol Song
College of Info. Science & Technology, Drexel University, Philadelphia, PA 19104, USA, mjones@immaculata.edu, song@drexel.edu

ABSTRACT

Software design is a complex activity. A successful designer requires knowledge and training in specific design techniques combined with practical experience. Designing a dimensional model is not an exception to this challenge. We present Dimensional Design Patterns (DDPs) that can be used for designing dimensional models. We present a metamodel of the DDP and show how to integrate them into Kimball's dimensional modeling design process.

INTRODUCTION

Research and experience suggest that software design is difficult and time consuming. Brooks discussed the difficulties involved in designing and implementing large systems when he stated: "plan to throw the first one away" (Frederick P. Brooks, 1995). The expertise for designing systems is acquired over a period of time by repeatedly performing design activities and refining the understanding of the system. The first system can be considered a prototype that builds knowledge and experience. Software engineers continually seek methods for improving the efficiency and effectiveness of the design process. Concentration on design techniques alone, however, has not guaranteed successful design solutions (Brooks, 1995). The challenge remains – how do software engineers obtain experience in order to create effective solutions?

Like other software design projects, data warehousing projects are complex, large, and difficult to design. A data warehouse is an integrated data repository specifically designed for performing analysis. The data for the data warehouse come from online transaction systems. That data is extracted, integrated, and stored in dimensional models. Business analysts perform tasks such as trend, time series, market, and risk analysis while data mining algorithms are used to uncover unknown information.

Several books provide data warehousing design strategies by documenting the dimensional models that solve specific data warehousing problems (Kimball and Ross, 2002; Silverston, 2001; Adamson and Venerable, 1998; Kimball, 1998). Organized by subject area, the referenced books describe and exemplify specific dimensional modeling techniques in relation to specific business applications. These books are intended to assist data warehousing practitioners in understanding data warehouse design by studying and learning from the examples of those more experienced in data warehouse design and implementation. Because the examples represent approaches to specific situations they may not address a given practitioner's particular design problem, but they are nonetheless valuable.

This research examines those dimensional models and identifies commonly occurring concepts and entities. These are referred to as Dimensional Design Patterns (DDPs). The idea to develop patterns for dimensional models resulted from 1) observing a variety of books on patterns in other software areas, 2) knowing there was a lack of detailed guidance for designing dimensional models, and 3) questioning whether patterns would be an appropriate mechanism for assisting data warehousing practitioners when creating dimensional models.

BACKGROUND

Data Warehousing and the Dimensional Model

Data warehousing databases are usually developed using a dimensional model. A dimensional model is a "specific discipline for modeling data that is an alternative to entity relationship modeling" (Kimball, 1998). Like an entity relationship model, a dimensional model reflects a data structure. However, a dimensional model is specifically designed to model data in a way that 1) emphasizes user understandability, 2) enhances query performance, and 3) accommodates change (Kimball, 1998; Corey, et al, 1998). To achieve these design characteristics, a dimensional model is typically denormalized.

A dimensional model is composed of two types of tables: 1) a fact table and 2) a dimension table. The schema associated with a dimensional model is referred to as a star schema since pictorially its structure resembles a star. The fact table is centrally located with the dimension tables "radiating" outward from the fact table. Fact tables contain foreign keys and measurements. Dimension tables represent and capture business entities used for analyzing the measurements. Dimension tables contain primary keys which associate the dimension attributes to the fact table, and textual descriptions which describe the attributes or characteristics of the business entities.

What is a Pattern?

The literature on patterns provides numerous definitions. As a result of studying those many definitions (Agerbo and Cornils, 1998; Alexander, 1979; Alexander et al., 1977; Buschmann et al., 1996; Coad, et al., 1995; Evitts, 2000; Fowler, 1997; Gamma, et al., 1995; Hay, 1996; Muller, 1999; Simon, 1996 and Vlissides, 1998), it becomes clear that the following characteristics are evident in those definitions.

- The problem that the pattern addresses is identified, recognized, and defined from real world situations.
- A pattern provides an approach for formulating a solution to a real world problem.
- The approach must be defined with respect to the real world context from which the problem emanates.
- The approach is reusable because it has been successfully used to solve recurring real world problems.
- A pattern endures over time.

In reviewing dimensional models, similarities in structure and content appear within, and in some cases, across subject areas. Since similarity is a prerequisite for pattern development it appears that pattern development for data warehousing applications is a promising research area.

DIMENSIONAL DESIGN PATTERNS (DDPS)

The purpose of the DDPs is assisting data warehousing practitioners when designing dimensional models by providing an approach for identifying dimensions in a systematic and usable way. When studying existing dimension tables, one begins to observe categories of repeated requirements that manifest themselves across the design of many dimensional models. Kimball's examples (Kimball and Ross, 2002) and Adamson and Venerable's examples (Adamson and Venerable's, 1998) were the basis for identifying the DDPs.

In order to create useful and usable DDPs the following criteria were adhered to:

- DDPs are explained via a commonly known and recognized mental model with the intent of increasing the practitioner's ability to understand, remember, and apply the DDPs.
- DDPs facilitate the identification of commonly used dimensions thereby providing a greater potential for improving design correctness with the initial model.
- DDPs are common across many dimensional models, thus reusability is improved and design time may be decreased.

DDP Mental Model

The mental model used for teaching students to write a story is the basis for DDPs. The writer considers the "who, what, when, where, and why" components of the story. Each component integrates various aspects of the tale in a creative yet structured way.

The characters involved in the story are created and developed when considering the "who" component. Stories usually refer to important entities and the ideas for those entities are generated when considering the "what" component. Every story is set within a particular time frame and the "when" component aids the writer in identifying the time period. The story location is of particular importance and the "where" component aids the writer in determining the story's setting. The motivation or the reasons behind the story are determined by considering the "why" aspects of the story.

When studying existing dimensional models, it becomes apparent that each model reflects the past activities of a business. Historical data can be assembled and queried in a way that tells a variety of stories. In some cases, business analysts may know or suspect those stories or they may uncover stories that have surprise endings. Therefore the telling of a story is the common mental model for identifying the potential dimensions for a dimensional model.

Domain Dimensions

In this subsection, we present the backgrounds and usage of each dimension pattern.

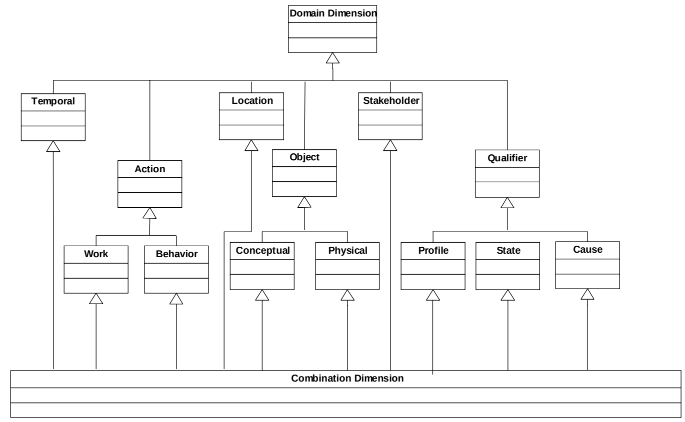
The Domain DDPs describes core subject area characteristics against which measurements are taken and questions are posed. DDPs (Figure 1) are classified as the following types: temporal (when), location (where), stakeholder (who), action (what is done or accomplished), object (what), and qualifier (why).

When performing analyses, it is important to know when events occur. Facts are historical and those facts need to be placed in the context of a time period in order for the software engineer to perform meaningful analysis. As observed in the dimensional models studied, a time period is typically described in terms of the traditional calendar year, fiscal calendar, time, and/or a special period. Therefore, the *temporal* DDP is represented as an aggregate of the traditional calendar year, fiscal calendar, time, and special period classes.

At times it may be necessary to place facts in the context of where they occur. Therefore, the obvious use of the location class is capturing a specific locale (e.g., region, state, city, township, etc.) for a facility (e.g., store, warehouse, etc.). However it may also be helpful to gain knowledge regarding the operational attributes of a facility. The *location* DDP class relationships are represented as an aggregate relationship which captures the location and operational attributes of a facility. More specifically, it captures the facility's purpose, individuals associated with managing the facility, locale, contact or communication data, and its physical (or configuration) characteristics.

A stakeholder can be associated with and described by an organization or a role. An organization is considered a group of people bound by

Figure 1. Domain DDP Class Diagram



common work, goals, or interests while a role describes a specific task and set of responsibilities. The *stakeholder* DDP describes individual stakeholders in terms of the organization to which they belong or their specific capacity or role they perform. It may be used for measuring aspects of an organization or a specific role. The designer needs to decide whether it is necessary to evaluate the performance of the organization and/or the specific roles within the organization.

An action is defined as accomplishing (or doing) something or exhibiting a behavior. For the action DDP there are two types of actions - work or behavior. Therefore the *action* DDP is modeled as a generalization-specialization. It is typical to measure characteristics associated with the work of accomplishing a specific task. For example, in a manufacturing scenario the analyst may need to evaluate the efficiency of production runs, a retail store may want to assess the effectiveness of a promotion, or a credit company may want to measure the timeliness of credit payments. In many industries it is important to evaluate the behavior associated with an action. For example, an organization may want to determine customer satisfaction via the number and type of complaints received through the customer service group.

The *object* DDP has two types of objects; conceptual and physical. A conceptual object is defined as an abstract idea used within the domain; whereas a physical object is defined as a tangible entity that is used within the domain. *Conceptual objects* may be various types of accounts such as revenue accounts, general ledger accounts, savings accounts, checking accounts and expense accounts. Also in the insurance industry, a policy that documents the insurance coverage is an example of a conceptual object. In education, a course may be thought of as conceptual object. Examples of *physical objects* are products, items, components (used to assemble products), and ingredients.

There are cases when information is more meaningful if it can be classified by a readily identifiable grouping attribute. The *qualifier* DDP classifies data by profile, state, or causal indicators. The *profile* DDP is used to group facts based on an identifying characteristic associated with a particular group. For example, employee skill groups can profile employees by their expertise with specific skills. The *state* DDP describes the circumstances characterizing a particular condition at a point in time. Associating a customer's satisfaction with regard to a product or service characterizes the customers' level of happiness (or state) at a point in time. The *cause* DDP is used to ascertain a possible reason for the occurrence of an event. A retail store is likely to be interested in the effectiveness of an advertising campaign. Therefore it is important to link the specific advertisement to the customer's purchases. Gathering and analyzing this data over a period of time may indicate the advertisement's influence on the customer's buying behavior.

When applying this model, the practitioner is not limited to using only a single class. A new dimension can be created by combining attributes from multiple classes to create a single *combination* dimension.

Figure 2. Auto Maker Sales Dimensional Model

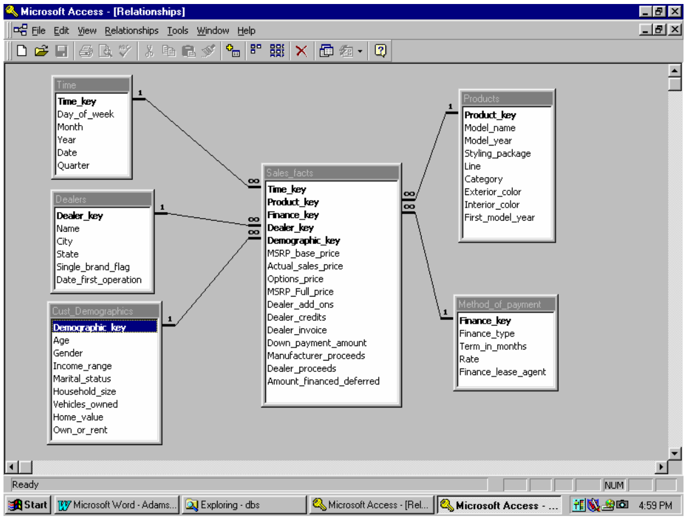


Table 1. Auto Sales Domain DDP Mapping

Auto Maker Sales Dimensions	***Dimension: DDP Mapping***	***Attribute: DDP Mapping***
Time	Temporal	Temporal – Calendar
Dealers	Stakeholder	Stakeholder – Organization Location – Locale Qualifier – Profile Temporal Calendar
Customer Demographics	Qualifier	Qualifier – Profile
Products	Object	Object – Physical Qualifier – Profile Temporal Calendar
Method of Payment	Action	Action – Work Qualifier – Profile Temporal – Calendar Stakeholder – Role

Mapping the DDPs to an Existing Model

In this section, we illustrate the usage of DDPs using auto maker sales dimensional model (Adamson and Venerable, 1998). This model (Figure 2) is composed of the sales fact table and the following dimension tables: *time, dealers, customer demographics, products*, and *method of payment*.

Applying the Domain DDP model to the Auto Make Sales example illustrates the use of all Domain DDP classes in a single dimensional model (Table 1). The column labeled ***Dimension: DDP Mapping*** represents the DDP class and the column labeled ***Attribute: DDP Mapping*** applies to the dimension table attributes. Multiple mappings in the ***Attribute: DDP Mapping*** column represent the application of combined dimension concept while a single mapping illustrates the use of one DDP class.

The following explains the Auto Sales DDP Mapping:

- The *time* dimension stores attributes that represent the traditional calendar structure.
- The *dealers* dimension represents the organization who sells the automobiles. The organization is described in terms of its city and state. The dealers can be profiled by using the single brand flag which indicates which dealers sell a single brand of automobile. The date of first operation indicates when the dealer first opened for business.
- All of the attributes in the *customer demographics* dimension provide the analyst with the ability to profile the dealers' customers.
- The *products* dimension represents the automobile. Its attributes describe ways to classify (styling package, line, category, exterior color, interior color) each automobile. The model year and first model year represent the temporal dimension.
- The *method of payment* dimension is the method in which the customer purchases the automobile. The term in months and rate attributes are profiling attributes while the finance lease agent is the individual who completed the lease.

CONCLUSIONS AND FUTURE WORKK

In this paper, we have presented the dimensional design patterns (DDPs) abstracted from case studies and examples presented in literature. Our DDPs cover temporal, action, location, object, stakeholder, qualifier, and combination. As an example of using the DDPs, we illustrated the mapping between our DDPs and a case study.

Our future work includes experiments and evaluation. A traditional pre-test post-test experiment will be conducted to determine the efficiency and effectiveness of the DDPs. The experiment will be conducted in an academic setting with gradate students studying information science. The experiment measures the time and correctness of design problems between the control group (no exposure to DDPs) and the experimental group (exposure to DDPs). In addition a post study evaluation will be used to compare and contrast the subjects' personal experiences with using the current method for creating dimensional models and the use of DDPs.

REFERENCES

Adamson, Christopher and Venerable, Michael. Data Warehouse Design Solutions. New York: John Wiley & Sons, Inc., 1998.

Agerbo, Ellen and Cornils, Aino. "*How to Preserve the Benefits of Design Patterns*", Proceedings of the Conference on Object - Oriented Programming, Systems, Languages, and Applications, October 18 – 22, 1998, Vancouver, Canada, pp. 134 – 143.

Alexander, Christopher; Ishikawa, Sara; Silverstein, Murray; Jacobson, Max; Kiksdahl-King, Ingrid; and Angel, Shlomo. A Pattern Language: Towns, Building, Construction. New York: Oxford University Press, 1977.

Alexander, Christopher. The Timeless Way of Building. New York: Oxford University Press, 1979.

Brooks, Frederick P. The Mythical Man-Month. Reading: Addison Wesley Longman, Inc., 1995.

Buschmann, Frank; Meunier, Regine; Rohnert, Hans; Sommerlad, Peter; and Stal, Michael. Pattern – Oriented Software Architecture: A System of Patterns. Chichester: John Wiley & Sons, 1996.

Coad, Peter; North, David; and Mayfield, Mark. Object Models – Strategies, Patterns, & Applications. Englewood Cliffs: Yourdon Press, 1995.

Corey, Michael, J.; Abbey, Michael; Abramson, Ian; and Taub, Ben. Oracle8 Data Warehousing – A Practical Guide to Successful Data Warehouse Analysis, Build, and Roll-Out. Berkeley: Osborne/ McGraw-Hill, 1998.

Evitts, Paul. A UML Pattern Language. Indianapolis: Macmillan Technical Publishing, 2000.

Fowler, Martin. Analysis Patterns: Reusable Object Models. Menlo Park: Addison-Wesley, 1997.

Gamma, Erich; Helm, Richard; Johnson, Ralph; and Vlissides, John. Design Patterns: Elements of Reusable Object-Oriented Software. Reading: Addison Wesley Longman, Inc., 1995.

Hay, David. Data Model Patterns: Conventions of Thought. New York: Dorset House Publishing, 1996.
Kimball, Ralph and Mertz, Richard. The Data Webhouse Lifecycle Toolkit. New York: John Wiley & Sons, Inc., 2000.
Kimball, Ralph; Reeves, Laura; Ross, Margy; and Thornthwaite, Warren. The Data Warehouse Lifecycle Toolkit. New York: John Wiley & Sons, Inc., 1998.
Kimball, Ralph; and Ross, Margy. The Data Warehouse Toolkit, 2nd ed. New York: John Wiley & Sons, Inc., 2002.
Muller, Robert. Database Design for Smarties: Using UML for Data Modeling. San Francisco: Morgan Kaufmann Publishers, Inc, 1999.
Silverston, Len. The Data Model Resource Book, Vol. 1, revised ed. New York: John Wiley & Sons, Inc., 2001.
Simon, Herbert A. The Sciences of the Artificial, 3rd ed. Cambridge: The MIT Press, 1996.
Vlissides, John. Pattern Hatching: Design Patterns Applied. Reading, MA: Addison Wesley Longman Inc., 1998.

Related Content