

Dimensional Modeling: Identifying, Classifying & Applying Patterns

Mary Elizabeth “M.E.” Jones
Drexel University
College of Information Science & Technology
Philadelphia, PA 19104
mjones@immaculata.edu

Il-Yeol Song
Drexel University
College of Information Science & Technology
Philadelphia, PA 19104
songiy@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 embodies this challenge. This paper presents Dimensional Design Patterns (DDPs) and their applications to the design of dimensional models. We describe a metamodel of the DDPs and show their integration into Kimball’s dimensional modeling design process so they can be applied to design problems using a known practice. By providing a metamodel and a method for DDP use, we combine theory and a practical design technique with the goal of increasing the efficiency and effectiveness of the software designer. The initial experimental results regarding the classroom use of DDPs revealed a significant increase in the efficiency of students to design a dimensional model, but more testing is necessary in order to evaluate the effectiveness measure.

Categories and Subject Descriptors

H.2.1 [Logical Design] *Data models*; D.2.10 [Design] *Methodologies*; D.2.0 [Software Engineering] *General - Standards*

General Terms

Design.

Keywords

Data warehouse, dimensional modeling, patterns, software engineering

1. 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” [5]. The expertise for designing systems is acquired over a period of time by repeatedly performing design activities and refining the understanding of the

system. Therefore, a first system can be considered a prototype that builds knowledge and experience. As a result, 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 [5]. The challenge remains – how do software engineers obtain experience in order to create effective solutions?

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.

Like other software design projects, data warehousing projects are complex, large, and difficult to design. Compounding the design difficulties is the lack of detailed guidance regarding dimensional modeling. Several books provide data warehousing design strategies by documenting the dimensional models that solve specific data warehousing problems [1, 13, 14, 15, 17]. 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.

Our research examines those dimensional models, identifies commonly occurring concepts and entities, and abstracts into dimensional patterns. This paper presents Dimensional Design Patterns (DDPs) and their applications to the design of dimensional models. The idea to develop DDPs 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, 3) questioning whether patterns would be an appropriate mechanism for assisting data warehousing practitioners when creating dimensional models, and 4) wanting to create a structured method that is both useful and useable when thinking about, discussing, determining, and reviewing dimensional models. The primary contributions of this paper are as follows. First, we present a taxonomy of dimensions, Dimensional Design Patterns, abstracted from over 50 case studies. Second, through a DDP example, we illustrate the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DOLAP’05, November 4–5, 2005, Bremen, Germany.
Copyright 2005 ACM 1-59593-162-7/05/0011...\$5.00.

mapping between the DDPs and a case study. Third, our initial experiments show that the DDPs significantly reduce the time to design dimensional models. The experimental group's time was reduced by 32.7% attributed through use of the DDPs and practice. The control group reduced their time by 27.3% through practice. Thus the experimental group had a 20% greater advantage in time than the control group after one week of instruction and practice. Our initial experimental results, however, reveal that more experimental data is required before drawing definitive conclusions on the correctness of the schema design.

The remainder of this paper is organized as follows: Section 2 provides background information regarding data warehousing, dimensional modeling, and software patterns. Section 3 (a) presents the driving concepts used to create the DDPs in order to increase DDP usefulness and usability, (b) defines and explains the Domain Dimension DDPs, (c) provides an example illustrating the applicability of DDPs, and (d) describes how to apply the DDPs when designing a dimensional model. Section 4 describes the experiment and also documents and discusses the initial results. Section 5 concludes the paper and presents the status of our on-going research.

2. BACKGROUND

2.1 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" [14]. 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 [8, 14]. 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.

2.2 What is a Pattern?

The literature on patterns provides numerous definitions. As a result of studying those many definitions [2, 3, 4, 6, 7, 9, 10, 11, 12, 16, 18, 19] 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.

3. 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 [15]) and Adamson and Venerable's examples [1] 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.

3.1 DDP Mental Model

Through research studies in human computer interaction designers recognize the importance of employing common mental models as design strategies for creating useful and usable interfaces. Software patterns can be difficult to learn, remember and apply. Therefore 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. 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 a reasonable mental model for identifying the potential dimensions for a dimensional model.

3.2 Domain Dimensions Explained

The Domain DDP Class Diagram (Figure 1) presents core dimensions presented in a class diagram against which measurements (facts) are taken and questions are posed. The Domain DDPs are classified as the following types: temporal (when), location (where), stakeholder (who), action (what is done or accomplished), object (what), and qualifier (why).

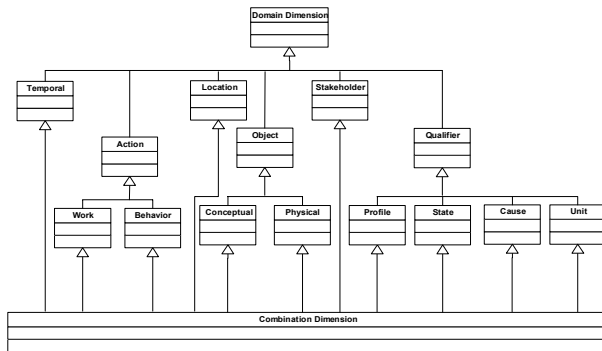


Figure 1. Domain DDP Class Diagram.

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 (Figure 2).

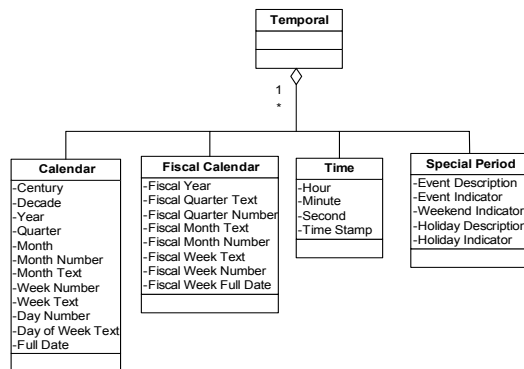


Figure 2. Temporal DDP.

At times it is 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 (Figure 3). 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.

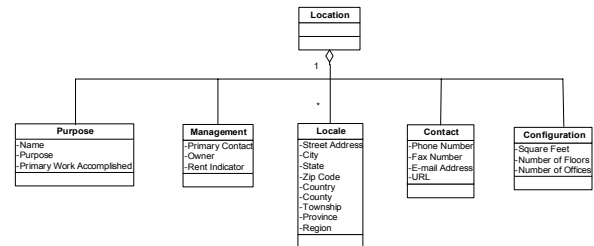


Figure 3. Location DDP.

A stakeholder can be associated with and described by an organization or a role. An organization is considered a group of people bound by 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 (Figure 4). 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.

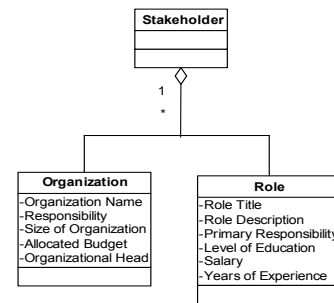


Figure 4. Stakeholder DDP.

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 (Figure 5). 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.

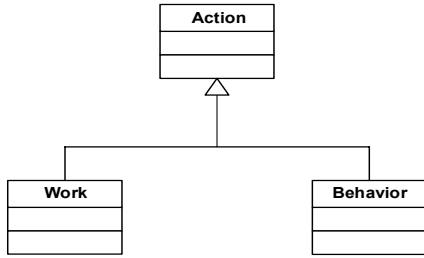


Figure 5. Action DDP.

The *object* DDP has two types of objects; conceptual and physical (Figure 6). 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.

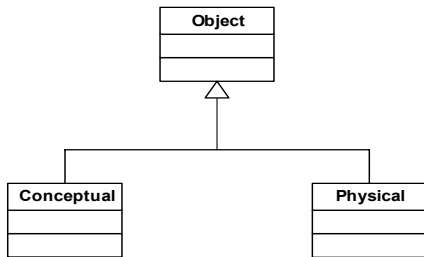


Figure 6. Object DDP.

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, causal, or unit indicators (Figure 7). 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. The *unit* DDPs is used to define measurement data in its associated unit of measure.

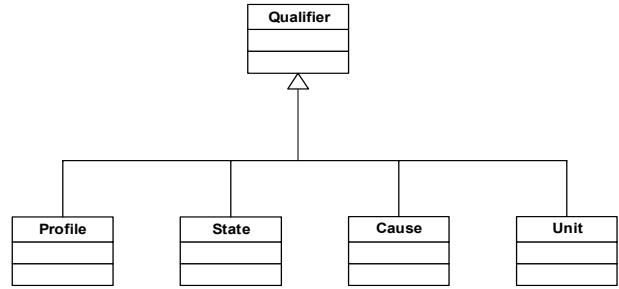


Figure 7. Qualifier DDP.

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 1).

3.3 Mapping the DDPs to an Existing Model

This section illustrates the applicability of DDPs by mapping the Domain DDPs to an existing dimensional model [1]. This model (Figure 8) 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 Domain DDP classes in a single dimensional model (Table 1). The column labeled **Dimension:** *DDP Mapping* represents the mapping of the dimension table to the DDP classes. The column labeled **Attribute:** *DDP Mapping* applies to the dimension table attributes and represents the mapping of dimension table attributes to the DDP classes.

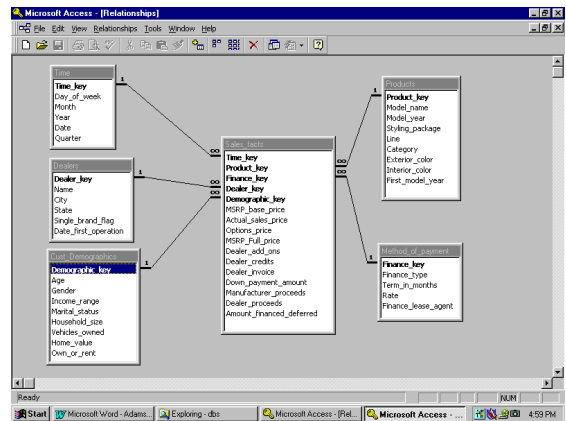


Figure 8. Auto Maker Sales Dimensional Model [1].

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 Qualifies - Status
Products	Object	Object – Physical Qualifier – Profile Temporal Calendar
Method of Payment	Action	Action – Work Qualifier – Profile Temporal - Calendar Stakeholder - Role

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.

3.4 Using Domain DDPs

The process for using the DDPs to design a dimensional model was created with simplicity in mind. In general, if a concept is cumbersome, it is unlikely that software engineers will adopt it for use. This process provides the framework for thinking about and determining the dimensions and attributes without the burden of an overly complex process.

When beginning a dimensional model, a business process must be selected and the grain of the data must also be determined. After those steps, the DDPs should be used to assist in determining the dimensions and attributes. The software engineer should be familiar with DDPs. A copy of the DDP class diagrams, their definitions and examples should be available for reference.

To begin determining the dimensions, the process requires systematically considering each Domain DDP individually in relation to the business process being modeled (Figure 9). It is critical to ensure each Domain DDP is evaluated in terms of the business process. Most likely the software engineer or business analyst will need to iterate through the process several times. At the conclusion of considering each DDP individually, the dimensions and attributes should be re-evaluated for accuracy. This provides an additional iteration through the dimensions to add, edit or delete items from the dimensional model.

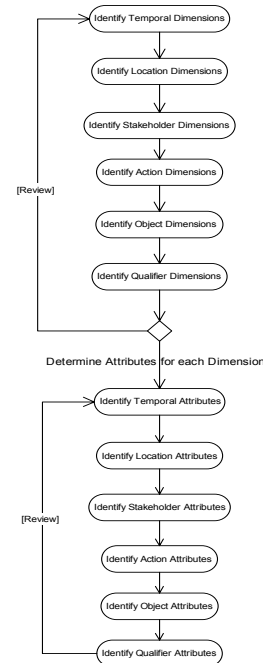


Figure 9. Domain DDP.

Table 2 and Table 3 should be used in conjunction with the activity diagram (Figure 9) to facilitate the identification of dimensions and attributes. It is the practitioner’s method for applying the Domain DDP class diagram. For each Domain DDP, the associated set of questions leads the analysts to consider and determine the specific dimensions and attributes for their problem.

Table 2. Domain DDP Dimension and Attribute Identification Questions.

<i>Temporal – Does the dimension need to:</i>	<i>Location - Does the dimension need to:</i>	<i>Stakeholder - Does the dimension need to:</i>
<ul style="list-style-type: none"> • Capture calendar periods? • Capture fiscal periods? • Capture both calendar and fiscal periods? • Capture special periods such as academic semesters or holiday seasons, etc.? • Capture time? 	<ul style="list-style-type: none"> • Capture a specific locale (site)? • Describe the purpose or work accomplished at that locale? • Capture significant dates/times associated with the location? • Associate stakeholders with the locale? • Describe the configuration of a locale? • Capture contact/communication information (e.g., phone, e-mail, etc.) for the locale? • Capture events or actions associated with the locale? • Associate the location with an object? 	<ul style="list-style-type: none"> • Describe the performance of an organization or a group? • Describe the performance of a specific individual or role of an individual? • Associate the stakeholder with a date and/or time? • Associate the stakeholder with a locale? • Associate the stakeholder with an event or action? • Associate the stakeholder with an object? • Describe the stakeholder with profile information? • Describe the stakeholder in terms of a state (e.g., single, married, etc.)?

Table 3. Domain DDP Dimension and Attribute Identification Questions.

<i>Action - Does the dimension need to:</i>	<i>Object (physical or conceptual) - Does the dimension need to:</i>	<i>Qualifier (profile, state, causal, unit)- Does the dimension need to:</i>
<ul style="list-style-type: none"> • Capture specific actions? What actions? • Identify types of work (steps) associated with the action? • Describe outcomes or results of work? • Identify or describe behaviors? • Associate a date/time with the actions? • Associate the action with a location? • Associate the action with stakeholders? • Associate the action with an object(s)? • Associate the action with profile information? • Associate the action with status information? • Associate the action with causal information? 	<ul style="list-style-type: none"> • Describe the classification of objects? • Associate the object with a date/time? • Describe the “life span” of the object? • Identify ownership of objects? • Associate the object with its location? • Associate the object with an action(s)? • Describe the configuration of the object? • Associate the object with profile information? • Associate the object with status information? • Associate the object with causal information? 	<ul style="list-style-type: none"> • Provide profile information of a stakeholder, location, action, or object? • Describe the status (or state) of a stakeholder, location, action, or object? • Describe causal situations associated with a stakeholder, location, action, or object? • Describe a measurement in terms of its unit of measure?

4. EXPERIMENT AND RESULTS

4.1 Experimental Procedure

A pre-test post-test experiment with an initial number of 21 undergraduate database students was conducted to determine the efficiency and effectiveness of the DDPs. Eighteen students completed the experiment. The experiment measured the time and precision/correctness of design problems between the control group (no exposure to DDPs) and the experimental group (exposure to DDPs). The subjects had an understanding of database concepts and system analysis, but minimal experience with data warehousing concepts. The following null hypotheses were tested to evaluate whether exposure to the

DDPs assisted undergraduate students in designing dimensional models.

- H₀: Dimensional design patterns (DDPs) have no impact on the time to design a dimensional model.
- H₀: Dimensional design patterns (DDPs) have no impact on the correctness of the dimensional model design.

The experiment was conducted in four steps (Figure 10). The experiment began with a pre-study questionnaire for capturing demographic data. In step 2, basic dimensional modeling concepts were taught to all subjects. The teaching was followed

by a dimensional modeling design problem to measure the subjects' initial understanding of those concepts. In step 3, the subjects were randomly assigned to either the control group (no exposure to DDPs) or the experimental group (exposure to DDPs). After teaching additional dimensional modeling and the DDP concepts, another design problem was administered to the subjects to evaluate the impact of the DDPs on the subjects' understanding of dimensional modeling. In the last step, subjects provided an assessment and comments regarding their experience with dimensional modeling examples and DDPs.

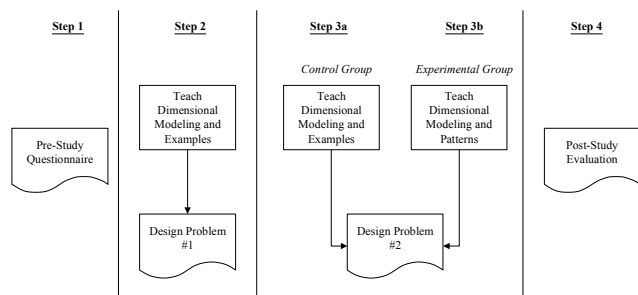


Figure 10. Experimental Procedure.

4.2 Experimental Statistics

Using SPSS, descriptive statistics and t-tests were generated to evaluate the time and correctness/precision data.

For the first design problem, subjects had a maximum of 60 minutes to complete the problem. The results of design problem #1 in terms of time were as follows.

	N	Min	Max	Mean	SD
Time – All Subjects	21	13	25	18.48	4.06

Design problem #1 was worth a total of 42 points. Scores were calculated as a ratio of the number of points earned divided by the total points. The results of design problem #1 in correctness/precision were as follows.

	N	Min	Max	Mean	SD
Correctness Precision – All Subjects	21	29	79	49.42	12.49

For design problem #2, the time variable was compared between the experimental and control groups. As in the first problem, subjects had a maximum of 60 minutes to complete their work. The following is a comparison of the two groups using descriptive statistics and a t-test.

	Mean	N	SD	Standard Error Mean
Time - Experimental Group	12.44	9	2.01	.67
Time - Control Group	13.44	9	1.51	.50

	<i>n</i>	<i>Correlation</i>	<i>Sig.</i>
Pair 1: Time - Experimental Group & Time Control Group	9	.958	.000

	Paired Differences					t	df	Sig (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
Pair 1: Time - Experimental Group Time Control Group	-1.00	.71	.24	-1.54	-.46	-4.243	8	.003

Design problem #2 was worth a total of 60 points. Scores were calculated as a ratio of the number of points earned divided by the total possible points (60). The results of design problem #2 in correctness/precision were as follows.

	Mean	N	Standard Deviation	Standard Error Mean
Precision - Experimental Group	35.7778	9	11.5734	3.8578
Precision - Control Group	39.1111	9	17.1861	5.7287

	n	Correlation	Sig.
Pair 1: Precision - Experimental Group & Precision Control Group	9	.489	.182

	Paired Differences					t	df	Sig (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
Pair 1: Precision - Experimental Group Precision Control Group	-3.3333	15.3216	5.1072	-15.1105	8.4439	-.653	8	.532

4.3 Experimental Results Discussion

The results of the first round of testing are promising. The efficiency measure (time variable) was significant ($p = 0.5$). The effectiveness measure (correctness/precision variable) was not significant and was most likely related to the number of participants completing the study ($n = 18$). The study will be repeated in August 2005 with an additional class and will hopefully yield results that can be combined with the current results to increase the significance.

Another factor influencing the usefulness of the current results is the presence of a score of 80% in the control group for the correctness/precision variable. The other scores in the group range from 23% to 47%. Incorporation of the 80% skewed the mean from 34% to 39% and resulted in the removal of the high score (80%) from the data set. By eliminating that data point the mean score of the experimental group was 34% compared to 36% for the control group.

These two implementations will increase the usefulness of the data and will provide stronger insight into the efficiency and effectiveness of the DDPs.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented the Domain DDPs abstracted from case studies and examples presented in the literature. Our DDPs cover temporal, action, location, object, stakeholder, qualifier, and combination patterns. Through a DDP example, we illustrated the mapping between the DDPs and a case study. We also provided a description for applying this process to a

design problem. Our initial experiment showed that the mean time for creating the initial dimensional model was 18.5 minutes and the mean time for creating a dimensional model using the DDPs was 12.4 minutes. Our initial experimental results, however, reveal that more experimental data is required before drawing definitive conclusions on the correctness/precision of the schema design. In addition, our on-going research includes the presentation of fact table and implementation patterns.

6. REFERENCES

- [1] Adamson, Christopher and Venerable, Michael. *Data Warehouse Design Solutions*. New York: John Wiley & Sons, Inc., 1998.
- [2] 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.
- [3] 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.
- [4] Alexander, Christopher. *The Timeless Way of Building*. New York: Oxford University Press, 1979.
- [5] Brooks, Frederick P. *The Mythical Man-Month*. Reading: Addison Wesley Longman, Inc., 1995.
- [6] Buschmann, Frank; Meunier, Regine; Rohnert, Hans; Sommerlad, Peter; and Stal, Michael. *Pattern - Oriented*

- Software Architecture: A System of Patterns*. Chichester: John Wiley & Sons, 1996.
- [7] Coad, Peter; North, David; and Mayfield, Mark. *Object Models – Strategies, Patterns, & Applications*. Englewood Cliffs: Yourdon Press, 1995.
- [8] 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.
- [9] Evitts, Paul. *A UML Pattern Language*. Indianapolis: Macmillan Technical Publishing, 2000.
- [10] Fowler, Martin. *Analysis Patterns: Reusable Object Models*. Menlo Park: Addison-Wesley, 1997.
- [11] Gamma, Erich; Helm, Richard; Johnson, Ralph; and Vlissides, John. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading: Addison Wesley Longman, Inc., 1995.
- [12] Hay, David. *Data Model Patterns: Conventions of Thought*. New York: Dorset House Publishing, 1996.
- [13] Kimball, Ralph and Mertz, Richard. *The Data Webhouse Lifecycle Toolkit*. New York: John Wiley & Sons, Inc., 2000.
- [14] Kimball, Ralph; Reeves, Laura; Ross, Margy; and Thornthwaite, Warren. *The Data Warehouse Lifecycle Toolkit*. New York: John Wiley & Sons, Inc., 1998.
- [15] Kimball, Ralph; and Ross, Margy. *The Data Warehouse Toolkit, 2nd ed.* New York: John Wiley & Sons, Inc., 2002.
- [16] Muller, Robert. *Database Design for Smarties: Using UML for Data Modeling*. San Francisco: Morgan Kaufmann Publishers, Inc, 1999.
- [17] Silverston, Len. *The Data Model Resource Book, Vol. 1, revised ed.* New York: John Wiley & Sons, Inc., 2001.
- [18] Simon, Herbert A. *The Sciences of the Artificial, 3rd ed.* Cambridge: The MIT Press, 1996.
- [19] Vlissides, John. *Pattern Hatching: Design Patterns Applied*. Reading: Addison Wesley Longman, Inc., 1998.