Triple-Driven Data Modeling Methodology in Data Warehousing: A Case Study

Yuhong Guo
Peking University - China
yhguo@pku.edu.cn

Shiwei Tang
Peking University - China
tsw@pku.edu.cn

Yunhai Tong
Peking University - China
yhtong@pku.edu.cn

Dongqing Yang
Peking University - China
dqyang@pku.edu.cn

ABSTRACT

In this paper, we present a useful data modeling methodology in data warehousing which integrates three existing approaches normally used in isolation: goal-driven, data-driven and user-driven. It comprises of four stages. Goal-driven stage produces subjects and KPIs(Key Performance Indicators) of main business fields. Data-driven stage produces subject oriented enterprise data schema. User-driven stage yields analytical requirements represented by measures and dimensions of each subject. Combination stage combines the triple-driven results. By triple-driven, we can get a more complete, more structured and more layered data model of a data warehouse. We illustrate each stage step by step using examples in our case study.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements / Specifications

General Terms: Design

Keywords

Requirement Analysis, Data Warehouse Design, Case Study

1. INTRODUCTION

Data warehousing has become one of the most important applications of database technology today. The new era of enterprise-wide systems integration and the growing needs towards business intelligence both accelerate the applications. Most large companies have established data warehouse systems as a component of information systems landscape [6].

One of the most important issues in data warehousing is how to develop appropriate data models to support querying, exploring, reporting and analysis. Although great achievements in research have been achieved on data warehousing, there is still a lack of comprehensive documentation and dissemination of requirement engineering methods [12], and related conceptual modeling is still under user's dissatisfactions [15]. Therefore, it is still important to research data modeling methodology in data warehousing.

Existing data warehouse development approaches can fall within three basic groups: data-driven, goal-driven and user-driven [10]. Each of the three approaches advocates only a single principle in data warehousing. Specifically, data-driven data modeling tries to construct data warehouse data models based only on operational system database schemas overlooking business goals and user needs. Goal-driven data modeling forms data models based only on business goals and accorded business processes ignoring data sources and user needs, and user-driven data modeling derives data models directly from user query requirements without considering data sources and business goals. Data models got from single principle are usually incomplete, which cannot obtain satisfaction and trust of organizations and individuals simultaneously. We hereby describe a triple-driven, multi level and integrated methodology for developing data warehouse data models based on our CLIC(China Life Insurance Company) case study, which provides a more complete, more structured and more layered data model of a data warehouse that organizations and individuals trust than working from a single principle.

Through the presentation of the methodology, the paper aims to tackle four research questions: how to integrate the three existing approaches to warehouse design; how to identify warehouse elements from operational data sources; how to embody corporate strategy and business objectives; and how to translate user requirements into appropriate design elements.

The remaining sections are organized as follows: Section 2 presents the related work. A brief background of the CLIC Data Warehouse Planning Project is given in section 3. Section 4 describes the proposed triple-driven methodology step by step. Finally, section 5 points out conclusions and future work.

2. RELATED WORK

Data-Driven Approaches: Data-driven approaches are widely used in different contexts [11, 19, 14, 13]. Data-driven data modeling in data warehousing starts with an analysis of transactional data sources in order to reengineer their logical data schemas. This raises two problems: (1) How to analyze transactional data sources and match them with information requirements to identify useful elements for data warehouse data models? (2) How to reorganize the identified source schema elements to form data warehouse data models based on the result of analysis? Although detailed data analysis and matching of data sources with business needs are important to data modelers, few literatures give concrete and systematic directions on them. Many researchers focus on the second problem. Dimensional models such as Star, Snowflake, StarER, ME/R are researched and widely used to reorganize data source schemas. In our paper, we focus on the first problem.

DOLAP'06, November 10, 2006, Arlington, Virginia, USA.
Copyright 2006 ACM 1-59593-530-4/06/0011…$5.00.

Goal-Driven Approaches: Goal-driven approaches place emphasis on the need to align data warehouse with corporate strategy and business objectives. By a review of literature, Rob Weir state that of the nineteen articles that referred to data warehouse implementations pre 2000, fifteen authors concluded that the 'Project must fit with corporate strategy and business objectives' [16]. Emerging reference architectures used in building enterprise data warehouse solutions are changing to meet business demands [18]. However, few articles present how to embody corporate strategy and business objectives to data warehouse data models. Böhnlein and Ulbrich-vom Ende present a representative goal-driven approach that is based on the SOM (Semantic Object Model) process modeling technique in order to derive the initial data warehouse structure [1]. However, this approach works only well when business processes are designed throughout the company and are combined with business goals. As a try, we focus on implementing business strategy in data warehouse data models by developing KPIs(Key Performance Indicators) of each business field and making up them to data models.

User-Driven Approaches: Like [3, 9, 17], user-driven approaches stress involvement of end users in data warehousing. Most of them mainly focus on requirement analysis process and deal with approaches facilitating user participations. In [9], the MD2 tool that aids users on identifying their analytical needs is presented. In [3], use cases are used for modeling user needs. In [17], a comprehensive method that supports the entire process of determining information requirements of data warehouse users is proposed. However, none of them focuses on how to translate user requirements into appropriate design elements.

A detailed comparison of data-, goal-, and user-driven can be found in [10]. This article concludes that the three methods are complementary and should be used in parallel to achieve optimal design. However, few literatures seem to address the integration of the three perspectives specifically. In [8], an Integrated-Process-Driven approach to data warehouse development is presented. The main idea is the integration of organizational processes and their respective data. This approach can be regarded as mixed data/goal driven. However, user-driven is not supported. The method in [5] can also be regarded as mixed driven. The main difference is that their method is top-down goal-driven centric, while ours is bottom-up data-driven centric. In our opinion, top-down design is difficult and can only capture very *limited* design elements. The approach in [2] is perhaps the closest to ours. It includes a top-down user-driven step, a bottom-up data-driven step and a final integration step. The main differences are: (1) their top-down step is based only on an informal and conversational approach, and report analysis, subject and KPIs techniques are not applied; (2) their bottom-up step needs a global integrated enterprise schema beforehand, while ours can start from several independent legacy system schemas; (3) their integration step gets the eventual model by taking intersection of the two steps results, while ours takes union of the three results aiming to capture elements adequately.

3. THE CLIC DW PLANNING PROJECT

In this section, we give a brief introduction of the CLIC(China Life Insurance Company) DW(Data Warehouse) Planning Project, which is used as our case study to clarify the full methodology.

CLIC is one of the largest insurance companies in China that deals with life insurance business. Just as the survey revealed in [7], internal needs and competitive pressure are the two critical factors that influence the top management to adopt data warehouse technology. They expect to build a central data warehouse that can not only centralize the data scattered in different operational systems, different departments and different areas; but also serve as a data base oriented with user querying, reporting, analysis and decision. In this context, the CLIC DW Planning Project was launched to develop data model of the central data warehouse.

There are 12 operational application systems involving business, finance, human resource, etc. As an example, we briefly introduce two systems, i.e. the Cbps system and the Callcenter system, which appear in later examples of the paper. The Cbps system is a core business process system, which supports the running of insurance business from customer applying, application checking, premium acceptance, commission payment to claim settlement. The Callcenter system is mainly used for customer consultation, complaint, inquiry, case reporting, etc. Both the systems record some aspects of customer characteristic and event information.

4. THE PROPOSED METHODOLOGY

Figure 1 illustrates the framework of our methodology. There are three stages: goal-driven stage, data-driven stage and user-driven stage. The goal-driven stage and the user-driven stage emphasize business issues, while the data-driven stage places emphasis on technical sides. Our methodology commences data modeling process with the goal-driven stage, followed by the data-driven stage and the user-driven stage in parallel. The eventual data warehouse's data schema is obtained with the subject-oriented enterprise data schema formed in the data-driven stage as a basis, making up the goal-driven KPIs and the user-driven analytical requirements represented by measures and dimensions.

Generally, the goal-driven stage covers requirements analysis and conceptual schema design phases. The data-driven stage covers detailed data analysis and logical data modeling phases. The user-driven stage covers requirements analysis and logical data model validation phases. The concept "*subject*" harmonizes the three stages and dominates the combination. Activities in each stage are described in the following subsections (4.1~4.4).

4.1 Goal-Driven Steps

Step 1.1: Develop Corporate Strategy. This step intends to ascertain enterprise's overall long-term goals and what measures the enterprise will take to achieve the goals.

Although corporate strategy is important to data warehousing, developing it is not easy and needs many human resources, especially high-level managers such as CEO, CFO, CIO; senior professionals; and market analysts. Based on their knowledge, rich experiences, and acute insight into market, overall development strategy, finance strategy, and market strategy are established. For example, many companies have developed their CRM strategies to retain long-term and profitable relationships with their customers. Data warehouse data modelers must be clear these strategies. The difficulty is that data warehouse team usually cannot mobilize senior managers to develop specific strategies for political reasons. Feasible ways include collecting corporate strategy and business objective information on the corporate web site, having fragmentary talks with senior mangers.

During the CLIC Data Warehouse Planning Project, three-layer strategies were specified in this step, namely, the corporate

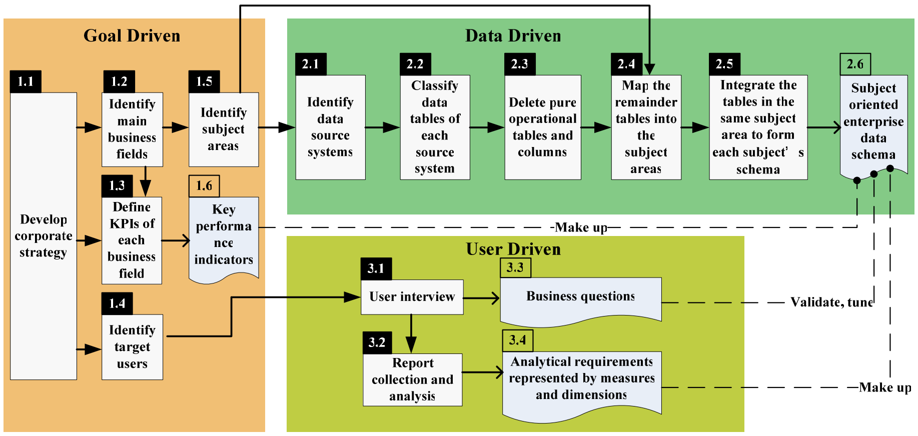

Goal Driven
Data Driven
User Driven
1.1
Develop corporate strategy
1.2
Identify main business fields
1.3
Define KPIs of each business field
1.4
Identify target users
1.5
Identify subject areas
1.6
Key performa nce indicators
2.1
Identify data source systems
2.2
Classify data tables of each source system
2.3
Delete pure operational tables and columns
2.4
Map the remainder tables into the subject areas
2.5
Integrate the tables in the same subject area to form each subject' s schema
2.6
Subject oriented enterprise data schema
3.1
User interview
3.2
Report collection and analysis
3.3
Business questions
3.4
Analytical requirements represented by measures and dimensions
Make up
Validate, tune
Make up

Figure 1. Framework of the proposed triple-driven methodology for DW design

strategy, the IT strategy, and the overall strategy and direction of the data warehouse. The IT strategy combines well with the corporate strategy, the strategy of the data warehouse surrounds with the IT strategy.

Step 1.2: Identify Main Business Fields. The objective of this step is to identify high-priority and high business return initiatives that data warehouse can support. These business fields are closely related to the overall strategy and direction of the data warehouse developed in step 1.1. More exactly speaking, these business fields are management topics to realize the corporate strategy.

During the CLIC Data Warehouse Planning Project, the four business fields are identified including CRM(Customer Relationship Management), RM(Risk Management), ALM(Asset Liabilities Management) and F&PM(Finance & Performance Management). The four business fields are applicable to most finance companies such as bank and insurance company.

Step 1.3: Define KPIs of Each Business Field. This step aims to define KPIs(Key Performance Indicators) of each business field. From these KPIs, we can determine attributes needed to support them in data warehouse data model.

Specific KPIs should relate to organization goals [4]. To define KPIs of each business field, goals, management techniques and challenges in each business field need to be investigated widely and deeply. One way is to use best practices of particular business field for reference. Then have regular meetings with business users and find out what metrics or measurements are important to them. The KPIs defined in this step will be mapped and made up to specific subjects during the combination stage in subsection 4.4.

Table 1 shows some general KPIs of CRM defined in the CLIC Data Warehouse Planning Project. More KPIs that are relevant to CRM can be found in [4]. Most of the KPIs are applicable to data warehouse design for companies actualizing CRM strategy.

Step 1.4: Identify Target Users. The aim of this step is to find who will use the data warehouse. Based on these users, decide who will be visited in the user-driven stage. Detailed user requirements will be collected in that stage.

To identify target users, organization tree structure, organization relationships and organization functions need to be investigated. This can be helped with human resource department. Generally, data warehouse users can be classified into four categories: query users, report users, analytical users and data miners in increasing complexity. Query users use data warehouse by simply executing SQL directly or indirectly. Report users get corresponding reports with a little more complicated application on the data which involves summarized computing. Analytical users focus on multidimensional analysis using OLAP tools, and data miners try to get knowledge by applying special algorithms on the data. The four class users penetrate through the organization tree and may overlap each other. Empirically, by analyzing organization function, main target users can be selected from organization tree.

Table 1. KPIs of Customer Relationship Management

KPIs	Definition
Customer Satisfaction Index	The quality of the services given by a department from the view of customers in the targeted segments.
Customer Retention Rate	The ability of a company's department to retain customers in the targeted measurement segments.
Revenue per Customer	The profitability on target customer segments.
Customer Acquisition Rate	The ability of a company's center/department to acquire customers in the target segments.

Step 1.5: Identify Subject Areas. The purpose of this step is to define what type of information (the so-called subject area, subject, or major information class) is required at a high level to conduct company's business, policies, procedures, and rules. Thus, data tables of source systems can be semantically mapped into the subject areas in order to organize data around the subject areas.

Identification of subject areas depends on business fields identified in step 1.2, modeling experiences and rich domain knowledge. The main approach is to determine what objects will be analyzed in each business field. Each subject can be seen as a business object, more abstract than entities described in logical data model. Detailedly speaking, "subject" can be seen as high-level information class of the whole information taxonomy of a data warehouse. So "subject" itself has levels, which means we can have "subject->sub-subject..." just like "country->province->city...". This is useful when there are so many information classes and one level classification is not enough. Empirically, the number of the subjects defined in each level is about 10, not more than 20, which is manageable for human. According to this guideline, "subject->sub-subject->entity" is enough for a large-scale data warehouse containing about 1000 tables, while "subject->entity" is enough for a medium-scale data warehouse with about 300 tables. Of course, this is not absolute in practice.

Examples of subjects in the CLIC Data Warehouse are like Customer, Claim, Policy, Channel, Campaign, Organization, Product, Risk rating, Asset, etc. Notice that "Claim" and "Campaign" correspond to event, while other subjects are like big dimensions. They are all big objects that users care for.

Deliverables 1.6: Key Performance Indicators. The main deliverables in the goal-driven stage, which will be directly taken to logical data model, are key performance indicators of each business field. The success of this stage depends greatly on the support of top management, as the entire organization is affected. That is, in order to quantify the strategy of the company and transfer the strategy into key performance indicators, senior managers, economists and data warehouse designers are required.

4.2 Data-Driven Steps

Step 2.1: Identify Data Source Systems. This step's purpose is to identify potential candidate data source systems of a data warehouse, whose data will probably be fed into a data warehouse.

Usually an enterprise has several application systems to support its business process, finance management, and human resource management. Not all the operational systems' data are valuable to a data warehouse based on specific data warehouse strategy and goals developed in the goal-driven stage. To determine candidate data sources, all the operational systems should be investigated. System functions, data interfaces, and database schemas are investigation emphases. Have a system as a candidate if any data elements may be valuable to any business fields or any subjects.

Step 2.2: Classify Data Tables of Each Source System. This step is to classify data source tables into the five categories:

1. *Transaction Tables (*see Transaction Entities in [11]). Transaction tables record details about particular events that occur in the business, e.g. insurance claims, salary payments. The key characteristics of a transaction table are: (1) It describes an event that happens at a point in time; (2) It contains measurements or quantities that may be summarized e.g. dollar amounts, weights, volumes.
2. *Component Tables* (see Component Entities in [11]). A component table is one that is directly related to a transaction table via a one-to-many relationship. It answers "who", "what", "when", "where", "how" and "why" of a business event.
3. *Report Tables*. Report tables record summary data about transaction tables and component tables. They exist in operational databases redundantly for efficiency.
4. *Classification Tables* (see Classification Entities in [11]). Classification tables are code tables related to component tables by a chain of one-to-many relationships, that is, they are dependent on a component table (directly or transitively).
5. *Control Tables*. Control tables record pure operational information used for operational process and control. USERS, ROLES, RIGHTS, SYSTEM PARAMETERS tables belong to this category. They are identified in this step and will be deleted as pure operational tables in the next step 2.3.

This classification approach for data tables we discuss here resembles the entity classification method mentioned in [11], which they use for developing dimensional models from traditional Entity Relationship models. Compared with [11], we add two new categories: report tables and control tables, as it is useful to differentiate the source data tables by the two categories in our method. Specifically, control tables are pure operational data tables without any use to data warehouse, while report tables represent summary information requirements of data warehouse.

Like [11], we define a precedence for resolving ambiguities when a table may fit into multiple categories: Transaction table (highest precedence) > Component table > Report table > Classification table > Control table (lowest precedence). For example, if a table can be classified as either a classification table or a component table, it should be classified as a component table. This is different from [11], in which Classification Entity has a higher precedence than Component Entity. We do this to make as many as tables valuable to analysis remain, as component tables are left and classification tables are ignored in next step. Notice: a preliminary normalization can be performed to reduce the ambiguities when several tables are at the same time transaction tables and component tables (as they are often not in 3NF in real databases).

Step 2.3: Delete Pure Operational Tables and Columns. This step aims to delete pure operational tables and columns that are meaningless to data warehouse.

First, delete the control tables identified in step 2.2. Second, keep the report tables and classification tables aside. The report tables will be considered together with the reports collected in step 3.2. The classification tables will be ignored in this step temporarily and be considered in physical data modeling phase, because the classification tables usually record attribute codes and corresponding attribute values that do not add new semantic information to data warehouse logical data model. The remainder tables are those transaction tables and component tables valuable to analysis. This is a relatively subjective process and whether a table is valuable to analysis is often controversial. Therefore, once it is difficult to make a decision, do not delete it. Finally, delete pure operational columns of the remainder tables such as "remarks". In our practice, over half of tables were deleted in the step. This reduced the workload in the latter steps greatly.

Step 2.4: Map the Remainder Tables into the Subject Areas. This step intends to map the remainder tables produced in step 2.3 into the subject areas produced in step 1.5 so that the data source tables with the same, similar, or interrelated business semantics can be easily integrated in the same subject area in step 2.5.

In fact, this step can be considered as further classification of the remainder transaction and component tables, which takes the subjects as "class labels". The difference is: in step 2.2 a table can only be classified into one category according to precedence hierarchy, while in this step a table may fall into two or more subjects. This is because a table may include two or more subjects' information, or the table is a link table among subjects. We call this process of classifying a table into more than one class

"Map". In our practice, Table 2 was used as the template to finish the process "Map". The actions of the process include:

- For each remainder table in each system, analyze which subject it belongs in to the greatest extent. Mark a "■"(filled_square) at the crossing of the table and the subject. The subject in which the table belongs to the greatest extent can be chosen by analyzing the "key" columns and the name of the tables, judging what object the table describes on earth. If it is difficult to choose among multiple candidates, for example, for a link table among multiple subjects, just pick one at will. This ensures each table has only one "■" in each line.
- Scan the remaining subjects. If the table relates to a subject, then mark a "□" (empty_square) at the crossing.

Notice: The "Map" process mentioned in this step is fit for "subject->entity" level in a medium-scale data warehouse. Refinement may be needed for "subject->sub-subject->entity" level in a large-scale data warehouse. The same is true of step 2.5.

Step 2.5: Integrate the Tables in the Same Subject Area to Form Each Subject's Schema. This step is to integrate tables mapped into the same subject to obtain each subject's schema.

There are two cases of integrating tables in the same subject. One is for component subjects; the other is for transaction subjects. By component subjects, we mean the subjects such as Customer and Policy, which have the characteristics as component tables defined in step 2.2. By transaction subjects, we mean the subjects like Campaign, Claim having the characteristics as transaction tables. As only transaction tables are mapped into transaction subjects, the integration for transaction subjects is relatively easy. The main work is to identify the transaction tables representing the same event in different systems, merge their columns to form one large table, and include all the central entities of the component subjects the event relates to as big dimensions. Notice: problems related to granularities, integrity constraints and codings being used for keys need be carefully considered when merging transaction tables that represent the same event but in different format or domain. In the following we only discuss integration for component subjects.

Usually a component subject is an analysis object that users care for. However, the information around the subject may scatter in different candidate operational systems. Step 2.4 gives an approach to gather data tables around each of subjects. After step 2.4, all the tables around a subject can be got by scanning the column grids of the subject in Table 2. The tables around a subject are those tables with a "■" or "□" at the crossed grids of the tables and the subject. Generally, there are the following four types of those tables around a component subject, according to the "■""□" mark and the class label (Transaction Table, Component Table):

1. *T■—Transaction tables with a "filled_square" mark.* They are usually the event tables recording an object's updating history. For example, the "UPD_Cust_Info" table, which records updating history of customer information, is such a table of "Customer" subject (see the second row in Figure 2(b)).
2. *C■—Component tables with a "filled_square" mark.* They record descriptive state information of a component subject. The "Cust_Info" table is such a table of "Customer" subject (see the third row in Figure 2(b).
3. *T□—Transaction tables with an "empty_square" mark.* They are transaction tables linking multi component subjects. These tables are mapped into transaction subjects with "■" and the multi component subjects with "□" (see the fourth row "Claim_Info" table in Figure 2(b)).
4. *C□—Component tables with an "empty_square" mark.* They are relationship tables between two or more subjects. "Policy_Beneficiary", which is a relation between "Customer" and "Policy", is such a "C□" table for "Customer" subject (see the fifth row "Policy_Beneficiary" table in Figure 2(b)).

Figure 2(a) shows the integration of the four types of tables in two systems to the same component subject. In the figure, the "T■" tables in each system record changing history of frequent update columns of the "C■" tables. What happens to the "T■" tables is not expressed in Figure 2(a) because they do not add new design elements to the schema of the subject. However, the data of "T■" tables will be loaded into "Sys1Sys2_C" during ETL phase by comparing timestamp. The "C■" tables in each system represent the same object. It is probable that multiple "C■" tables appear in one system. They are all integrated into the "Sys1Sys2_C" entity by merging columns of each "C■" table while deleting repetitive columns, like "joining" multiple tables to form one big view. The "T□" tables which usually record different events of the same object are taken directly to form different entities (shown as "Sys2_T, …, Sys1_T") with a one-to-many relationship with the "Sys1Sys2_C" entity respectively. The "C□" tables need not to be integrated, as they belong to other subjects ("Subject M" "Subject N"). However, as they are relationship tables between "Subject 1" and "Subject M", "Subject N", two one-to-many relationships are added from "Sys1Sys2_C" to the two entities in "Subject M" and "Subject N" respectively.

Deliverable 2.6: Subject Oriented Enterprise Data Schema. The main deliverable in the data-driven stage is the subject oriented enterprise data schema, which is composed of multiple integrated schemas of each subject formed in step 2.5.

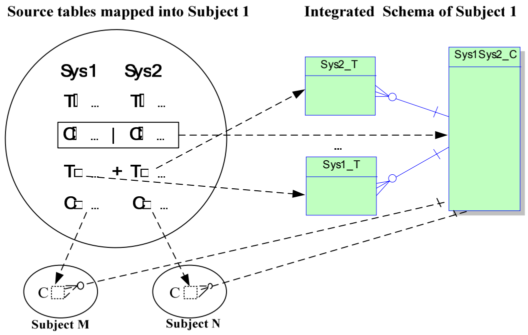
As an example, data schemas of a simplified "Customer" subject and a "Claim" subject from our CLIC data warehouse planning practice are given in Figure 3. The "Customer" subject as a component subject integrated the two system's customer related tables (the tables mapped into "Customer" subject in Figure 2(b)), forming a component entity "CbpsCallcenter_Customer" and two transaction entities "Cbps_Claim" "Callcenter_Consult". The "Claim" subject as a transaction subject was formed having the "Cbps_Claim" as its central fact entity and other component entities as its dimensions.

As we said above, there are two types of subjects: component subjects and transaction subjects. Each component subject has a central component entity recording descriptive status information of that subject, and some transaction entities recording different events of the central component entity. Each transaction subject has a central transaction entity recording detailed information about the related event, and some component entities around it as its big dimensions. This means each transaction subject's schema is a star schema with a central transaction fact entity and some component dimension entities. On the contrary, each component subject's schema can be seen as an *anti-star* schema with a central component entity and some transaction entities around it.

Compared with star schema, *anti-star* schema tries to collect all behavior related information of a component object together, instead of collecting all character related information of an event object together. It shifts its attention to a component object, not an event, which leads to delicate particular analysis of a specific object according to all its behaviors. This helps to find correlations among different events. For example, customer frauds may be found by comparing and tracing all their behavior events including claiming, consulting, changing passwords, and so on.

Table 2. Work template of step 2.4

Candidate Systems	Remainder Tables	Subject 1	Subject 2	…	Subject N
System1	Transaction Table1	■	□		
	Component Table1	■	□		
	…				
System2	Transaction Table1	□	■		□
	Component Table1	□	■		
	…				
…					


Source tables mapped into Subject 1
Integrated Schema of Subject 1
Sys1
Sys2
Sys2_T
Sys1Sys2_C
Sys1_T
Subject M
Subject N

		Customer	Claim	Policy
Cbps	UPD_Cust_Info	T■		
	Cust_Info	C■		
	Claim_Info	T□	T■	T□
	Policy_Beneficiary	C□		C■
	Clerk_Info			
	Policy_Info			C■
Callcenter	Customer	C■		
	Consult	T□		
	Clerk			

Figure 2. (a) Integrate tables in the same subject (b) Example of mapping tables into subjects

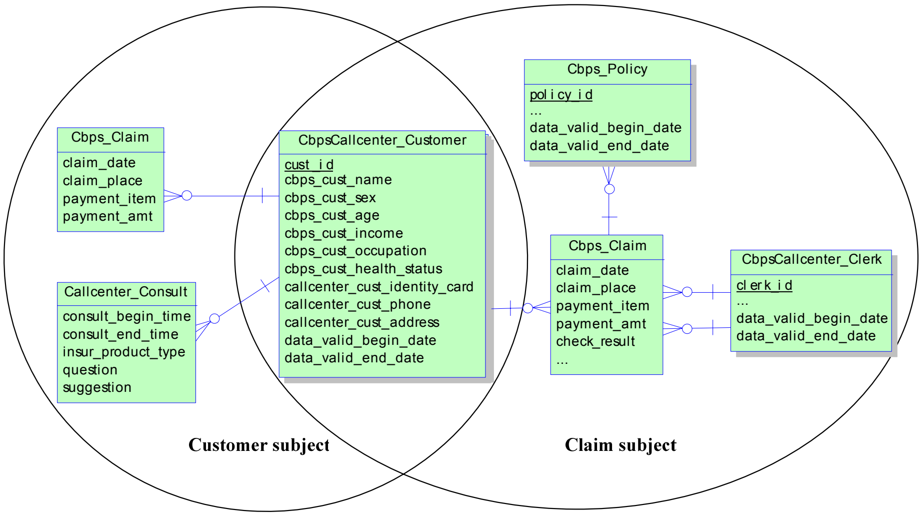


Figure 3. Customer subject (anti-star) vs. Claim subject (star)

4.3 User-Driven Steps

Step 3.1: User Interview. This step is to collect requirements of the target users identified in step 1.4 by interviewing.

Usually, user interview can be arranged department by department. The target department managers are especially needed to take this interview, as they have more comprehensive and deeper understandings about the department. In the interview, data warehouse requirements developers need introducing related data warehouse project context to interviewees in the beginning. Then discuss the questions prepared which are designed to help the interviewers induce the users to think out the requirements.

Step 3.2: Reports Collection and Analysis. This step aims to collect reports department by department after the user interview and analyze them to form analytical requirements. We do this because reports represent more concise, polished and refined data that users pay attention to than data in the operational database.

Reports can be divided into two types for a department. One is the report provided for the department from the lower organizations or other departments, the other is the report the department supplies for the upper organizations or other departments. By analyzing these two types of reports, detailed and comprehensive data requirements of the department are got. The report tables identified in step 2.2 should be included in the reports collected in this step.

Deliverables 3.3: Business Questions. By user interview, users' critical hot requirements can be probed and business questions expected to be answered with data warehouse can be abstracted, classified and prioritized in deliverable documents.

The business questions can be used to evaluate and validate data warehouse's logical data model. Generally, logical data model of data warehouse should try its best to answer these business questions so that users can accept it favorably. Typical business questions are like: Which customers are most profitable based upon premium revenue? Which channels customers like most? What are the top five reasons that customers return products?

Deliverables 3.4: Analytical Requirements Represented by Measures and Dimensions. By user interview, reports collection and analysis, analytical requirements represented by measures and dimensions for each subject can be obtained. The measures and dimensions should be complemented to the subject-oriented data schema produced in the data-driven stage. Figure 4 shows the analytical requirements represented by measures and dimensions for "Customer" subject. Notice: although "Customer" is not a so-called fact, it can have measures. This is because the so-called "fact" and "dimension" are relative. When we shift our attention to customers and try to observe them from different aspects, "Customer" becomes our focus and fact.

4.4 Combine the Triple-Driven Results

The last stage is to combine the triple-driven results formed above: the KPIs formed in the goal-driven stage (Deliverables 1.6 in subsection 4.1); the subject oriented enterprise data schema formed in the data-driven stage (Deliverable 2.6 in subsection 4.2); the analytical requirements represented by measures and dimensions formed in the user-driven stage (Deliverables 3.4 in subsection 4.3). The eventual result is a complete, subject-oriented logical data model of a data warehouse. The combination is based on the subject oriented data schema formed in the data-driven stage, making up the goal-driven and the user-driven results.

Figure 5 shows combining the triple-driven results to form the eventual data model of "Customer" subject. There are three parts in the logical data model of "Customer" subject. One is the part at the down left corner with each entity a solid line frame. This part is mainly obtained from the data-driven result shown in Figure 3 of deliverable 2.6. The second part is the goal-driven part at the top with each entity a dotted line frame, which is composed of "Customer_KPIs" entity and "Customer_Segment" entity to which the KPIs apply. The attributes of "Customer_KPIs" are composed of the KPIs (like KPIs shown in Table 1) defined in

Measures: count, premium, insured amount, policy numbers, net profit, loss ratio…

Dimensions: sex (male, female, unknown)
income (<1000, 1000-5000, 5000-10000, 10000-20000, >20000)
marriage status (married, unmarried, divorced)
education degree (<primary school, high school, >undergraduate)
age (< 20, 21-25, 26-30, 30-35, 36-40, 40-50, 50-65, >65)

Figure 4. Measures and dimensions for Customer subject

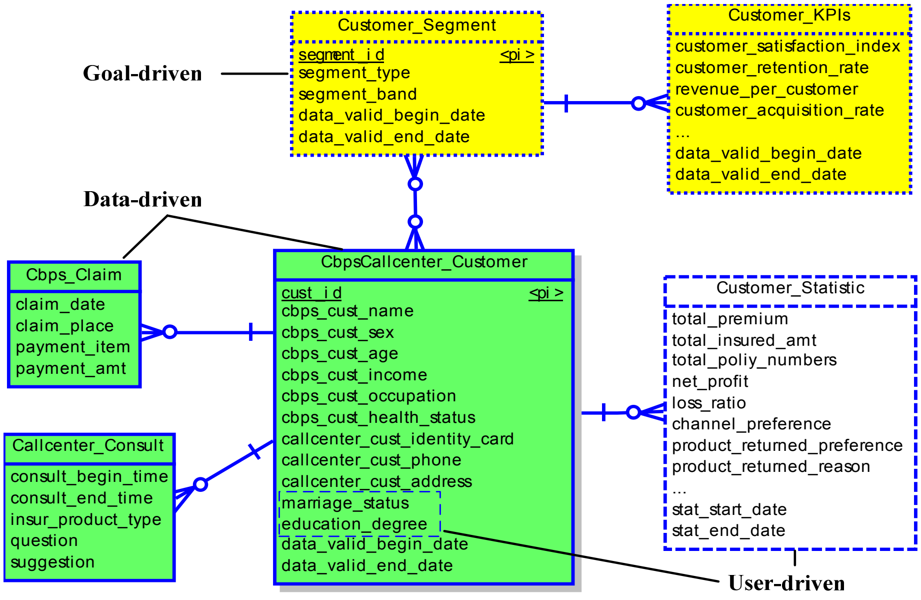


Figure 5. Combine the triple-driven results to form eventual data model of Customer subject

step 1.3 that belong to "Customer" subject. "Customer_Segment" entity is arrived at from the definitions of the KPIs in Table 1, which indicate the granularity the KPIs apply to is *customer segment* not a customer. The third part is the user-driven part at the down right corner with "Customer_Statistic" entity, which records statistic indexes of customers. The top five statistic indexes correspond to the measures of "Customer" subject in deliverables 3.4 (see Figure 4), and "channel_preference" "product_returned_preference" "product_returned_reason" are arrive at from the business questions listed in deliverables 3.3. Two dimensions (marriage_status, education_degree) are made up to the "CbpsCallcenter_Customer" entity as attributes.

In fact, the three parts above represent three different data layers of the same subject. The bottom layer is base data layer of the data-driven result part, which holds basic, crude, and atomic data collected from the operational systems. The medium layer is summary data layer of the user-driven result part, which holds aggregate, statistical data around a subject. The high layer is synthesis data layer of the goal-driven result part, which holds highly synthesized, deep computed data of a subject that high managers and decision-makers pay attention to. The three data layers are complementary each other, providing a relatively complete data view of a subject.

5. CONCLUSIONS

We have described a triple-driven methodology for developing data warehouse logical data model based on the CLIC case study. There are four stages of the methodology: (1) goal-driven stage, (2) data-driven stage, (3) user-driven stage, and (4) combination stage. The goal-driven stage produces subjects and KPIs of main business fields. The data-driven stage obtains subject oriented data schema. The user-driven stage yields business questions and analytical requirements. The combination stage combines the triple-driven results. The advantages of this methodology are:

- It ensures that the data warehouse reflects the enterprise's long-term strategic goals and accordingly ensures actual business value of the data warehouse as well as stability of data model furthest, which meets senior managers' expectations.
- It raises acceptance and trust of users towards the data warehouse with users' involvements in the user-driven stage.
- It ensures that the data warehouse is flexible enough to support the widest range of analysis, by including the three different data layers: the data-driven base data layer, the user-driven summary data layer, and the goal-driven synthesis data layer.
- It leads to a design capturing all specifications.

The impact of the methodology on our case study and the experience we have gathered by applying the method in the case study are encouraging. In the case study, we started from a situation where operational databases were scattered and not integrated, and business needs and users' requirements were ambiguous. The proposed method was indeed essential to direct us toward a solution that is both established in the data and oriented to business needs. The users' feedback was very positive: the design is effective and comprehensive, which can satisfy their different application requirements from user querying, reporting, to multidimensional analysis and management decision.

In the future, we intend to automate some steps of our methodology. Besides, further validation is needed, especially redundancy check and implementation evaluation.

6. REFERENCES

[1] Boehnlein, M. and Ulbrich vom Ende, A. Business process oriented development of data warehouse structures. In *Proc. Data Warehousing 2000*, Physica Verlag (2000)

[2] Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., and Paraboschi, S. Designing data marts for data warehouses. *ACM Trans. Softw. Eng. Methodol.* (2001) 10(4): 452-483

[3] Bruckner, R., List, B., and Schiefer, J. Developing requirements for data warehouse systems with use cases. In *Proc. AMCIS* (2001) 329-335

[4] Cunningham, C., Song, I.Y., and Chen, P. Data warehouse design to support customer relationship management analyses. In *Proc. DOLAP* (2004) 14-22

[5] Giorgini, P., Rizzi, S., and Garzetti, M. Goal-oriented requirement analysis for data warehouse design. In *Proc. DOLAP* (2005) 47-56

[6] Herrmann, C. and Melchert, F. Sponsorship models for data warehousing: two case studies. In *Proc. AMCIS* (2004)

[7] Hwang, H.-G., Kua C.-Y., Yen, D., and Cheng, C.-C. Critical factors influencing the adoption of data warehouse technology: A study of the banking industry in Taiwan. *Decision Support Systems, vol. 37, no. 1* (2004) 1-21

[8] Kaldeich, C. and Sa, J.O.e. Data warehouse methodology: A process driven approach. In *Proc. CAiSE* (2004)

[9] Laender, A., Freitas, G., and Campos, M. MD2 – Getting users involved in the development of data warehouse applications. In *Proc. CAiSE* (2002)

[10] List B., Bruckner R., Machaczek K., and Schiefer, J. A comparison of data warehouse development methodologies: Case study of the process warehouse. In *Proc. DEXA* (2002)

[11] Moody, D. and Kortnik, M. From enterprise models to dimensional models: A methodology for data warehouse and data mart design. In *Proc. DMDW* (2000)

[12] Nguyen, T.M., Min Tjoa, A., and Trujillo, J. Data warehousing and knowledge discovery: A chronological view of research challenges. In *Proc. DaWaK* (2005)

[13] Peralta, V., Illarze, A., and Ruggia, R. On the applicability of rules to automate data warehouse logical design. In *Proc. DSE workshop in CAiSE* (2003)

[14] Phipps, C., Davis, K. Automating data warehouse conceptual schema design and evaluation. In *Proc. DMDW* (2002)

[15] Rizzi, S. Open problems in data warehousing: 8 years later. In *Proc. DMDW* (2003)

[16] Weir, R., Peng, T., and Kerridge, J. Best practice for implementing a data warehouse: A review for strategic alignment. In *Proc. DMDW* (2003)

[17] Winter, R. and Strauch, B. Demand-driven information requirements analysis in data warehousing. *Journal of Data Warehousing* (2003) 38-47

[18] WilliamO'Connell. Trends in data warehousing: A practitioner's view. In *Proc. VLDB* (2004)

[19] Yang, L., Miller, R., Haas, L., and Fagin, R. Data-driven understanding and refinement of schema mappings. In *Proc. ACM SIGMOD* (2001)