Universe Scientific Publishing

ORIGINAL ARTICLE

# A Data Warehouse Design Method

**Yameng Ban*, Zhijuan Wang**

Shijiazhuang Information Engineering Vocational College, Shijiazhuang 050000, Hebei, China. E-mail: banyameng@163.com

***Abstract:***Data warehouse is an important part of data driven decision support system. The design of data warehouse is a complex task, which needs systematic method. This paper presents a design method of data warehouse based on UML, which spans three design stages: concept, logic and physics. Our approach consists of a set of meta models used in each phase, and a set of transformations that can be semi automated. Following our idea of object-oriented, we use UML to represent all meta models, and illustrate the formal specification of OMG based transformation of object constraint language.

***Keywords:***Data Warehouse; Decision Support;Meta Model

Data warehouse; decision support; meta model data warehouse has become an important part of organizational decision support. It has been proved that users can improve decision-making performance by implementing data warehouse. In order to gain business insight from the data stored in the data warehouse, decision makers usually use OLAP, query and report or data mining tools. OLAP is a multi-dimensional system model. The model provides managers with a business oriented data view. It helps data navigation, analysis and final decision-making. The design process of data warehouse is very important and needs to be supported by appropriate methods. An effective design method of data warehouse can not only ensure the quality of data warehouse, but also meet the changing needs of the environment or decision-makers, and it is also very important for the frequent evolution of data warehouse.

## 1. Related research

The design method of data warehouse can be analyzed from three aspects. First is related to OLAP tools. Suppliers of these tools claim that cube design (multidimensional modeling) is an intuitive and quasi immediate process, and does not require complex design methods. The second is to adapt to the classic database design method. Third, it is related to multiple source integration methods, in which data warehouse mode is the result of source database mode integration.

Open models or methods usually contain only certain stages of data warehouse design. These previous research work can be described according to the following standards: relevant examples, underlying conceptual models, abstraction levels covered by methods, system components, such as data warehouse, data mart, extract transform load (ETL) process, etc.

In addition, we can describe the design method as: it combines the top-down method with the bottom-up method. It is based on UML notation and promotes processes by referring to a well-known formalism. It is implemented independently by providing the rules of MOLAP and ROLAP tools. Standardization relies on the combination of UML and its associated OCL formalism. Formalization is based on a metamodel that describes three levels of abstraction (conceptual, logical, and physical). Transformation rules are provided to map one level to the next, and are formalized using OCL notation.



doi: 10.18686/ahe.v5i3.3478

## 2. The Data warehouse design method

The design method of data warehouse considers the user's requirements and the operation data source of data warehouse development. Data warehouse designers should first determine the information needed by decision makers, which is the only way to prevent key business requirements from being ignored. Starting from the informal statement of the information needed by the decision maker, the method is divided into concept stage, logic stage and physical stage. The following details the conceptual, logical, and physical design phases, i.e., the metamodels used and the transformations performed in each phase.

## 3. Conceptual design

In the conceptual design phase, we need a form of conceptual modeling to specify the user's information needs after confirming that the multi-dimensional meta model belongs to the logical layer, and then carry out the logical design. This form of conceptual modeling needs to be familiar to users and designers, such as ER or UML. In order to simplify the automatic mapping from conceptual model to multidimensional schema, we need to add some information specific to multidimensional modeling into the conceptual model. However, we should avoid including too much multi-dimensional information in the conceptual model.

The conceptual design stage is divided into two steps. The first step is to get a UML model, more accurately, a class diagram without operation. The second step enriches and transforms the model so that it automatically maps to a unified multidimensional schema. Four types of transformation are carried out: the determination of recognition attributes, the determination of measurement, the transfer of associated attributes and the transformation of generalization.

Using any requirement engineering method conforming to UML, the designer defines a UML class diagram to represent the initial information requirements of the decision maker. The first step of data warehouse design method does not use multi-dimensional concept, so it can maximize the reuse of system analysis methods commonly used in transactional systems engineering.

The second step of conceptual design is to promote the subsequent mapping from UML conceptual model to logical multidimensional pattern. To achieve this, we need to extend the standard UML metamodel and add as few concepts as possible to simplify automatic mapping. In order to extend UML, we use the extension mechanism of prototype and tag value unit.

## 4. Logical design

In the logic design phase, the rich and transformed UML conceptual model is mapped to the logic pattern represented by the concept of unified multidimensional meta model. Logical multidimensional patterns are generated through the transformations detailed below. The gradual application of the transformation is illustrated by an example.

Logic design is divided into five steps, each of which describes the results of the previous steps in detail:

(1) The definition of facts (transform tcl 1a and tcl 1b). The definition of fact includes its measurement and the definition of its related dimension level.

(2) Definition of hierarchy (transform tcl 2).

(3) Dimension definition.

(4) Definition of dimension level attribute (transform tcl 4).

(5) Definition of aggregation functions that can be applied to hierarchy metrics (transform tcl 5).

In our design approach, the applicable aggregation function is determined for each hierarchy in each dimension of each metric. In this way, we can accurately define the applicable aggregation function and specify when the aggregation function is only applied to the first N levels of the hierarchy. Our proposal goes beyond the traditional distinction between additive, semi additive and non additive facts. The latter distinction focuses on sum functions, and there is no clear distinction between hierarchies within a given dimension. In our method, we can say that a measure is additive if the sum function can be used along all dimension levels of all hierarchies of all dimensions of the measure.

## 5. Physical design

This phase maps the multidimensional schema into a physical database schema. Thephysical schema depends on the target MOLAP or ROLAP tool. So, a set of transformations is defined for eachtype of tool. We have defined transformations for

ROLAP star implementation. In this paper, we consider Oracle MOLAP, which is representative of the MOLAP tool category. We present the Oracle MOLAP metamodeland then the transformations that map a logical multidimensional schema into a physical Oracle MOLAP schema.

The Oracle MOLAP package is defined as a non-normative extension to the common warehousemetamodel. The metamodelis an adapted version, with the main concepts of the Oracle MOLAP tool. Physical Oracle MOLAP dimensions are the equivalent of logical dimension levels. Variable s correspond to logicalmeasures or dimension level attributes, while logical hierarchies are implemented by means of relations. Bothvariablesand relations are dimensioned. The dimensions of a variable or a relation are specified between "<>" in the definition ofthe variable or relation. A relation is a functional dependency between its dimension and its reference dimension. Dimensions are temporal or non-temporal. Possible types for non-temporal dimensions are ID, Text, and Integer. Possible types for temporal dimensions are Day, Week, Month, Quarter andYear. For variables, types are Boolean, Date, Decimal, ID, Integer, Short decimal, Short integer and Text.

A logical fact with only minimal dimensions cannot be mapped into an Oracle MOLAP relation. Consequently, for such a fact, we define a dummy Boolean variable in Oracle MOLAP. For each set of values of the dimension levels dimensioning the fact, the corresponding instance of the dummy variable will indicate whether the instance of the fact exists or not. Note that if a logical fact has at least one measure, this measure has been mapped into an Oracle MOLAP variable by transformation Tle2. If the measure is always defined, the variable is defined whenever the fact is defined. Therefore, in this case, a dummy variable is not necessary.

Thanks to these transformations, the definition of the Oracle MOLAP schema in the Oracle MOLAP command language can be generated from the logical multidimensional schema, in a quasi automated way. The generation of dimensions is straightforward. The process also generates new variables, going beyond the explicit requirements, such as the variable gets which materializes the relationship between regions and medias. This example illustrates the richness of the conceptual modeling aspects. It would not have been easy to obtain this variable without relying on the conceptual modeling process. Moreover, the choice of implementing the dimension level all results in a list of relations.

## 6. Conclusion

We describe a comprehensive data warehouse design method based on UML. On the basis of previous studies, we define a unified multidimensional meta model. The meta model is used to represent the logical multidimensional pattern in the design method of data warehouse. The unified multidimensional meta model can be used in the context of MOLAP or ROLAP tools, so as to provide standard and advanced views of data warehouse for decision makers. In the future, we still need to study the richness of mapping transformation, dealing with non data oriented requirements and tracking the mapping between different design stages.

## References


1. The OLAP Report – Market share analysis; 2008. http://www. olapreport.com/Market.htm.
2. Bonifati A, Cattaneo F, Ceri S, *et al*. Designing data marts for data warehouses. ACM Transactions on Software Engineering and Methodology 2007; 10 (4): 452-483.
3. Arkhipenkov S, Golubev D. Oracle express OLAP. Charles River Media; 2006.
4. Peralta V, Ruggia R. Using design guidelines to improve data warehouse logical design. 5th International Workshop on Design and Management of Data Warehouses; 2007.
5. Jarke M, Lenzerini M. Vassiliou Y, *et al*. Fundamentals of data warehouses, Second ed. Berlin: Springer-Verlag; 2008.
6. Kimball R. The data warehouse toolkit. JohnWiley&Sons; 1996.
7. Luján-Mora S, Trujillo J. A comprehensive method for datawarehouse design. 5th International Workshop on Design and Management of Data Warehouses; 2007.