ISSN No. 0976-5697

Volume 2, No. 2, Mar-Apr 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Object Oriented Approach using Multidimensional OLAP Data Model

Alankrita Aggarwal*
Assistant Professor, CSE
H.C.T.M, Kaithal ,India
alankrita.agg@gmail.com

Meenakshi Sharma
Senior Lecturer,CSE
H.C.T.M, kaithal ,India
minny_k@gmail.com

Abstract: The introduction of metadata management concept is given which is based on the object-oriented paradigm for modelling and querying multidimensional OLAP-data. In this model we apply the basic concepts of the object-oriented model including object, class, and relationship between objects to describe objects of multidimensional data and the OLAP operations. In our approach the multidimensional OLAP-data is modelled as a fact and a set of dimensions that are organized into class-hierarchy. On-line Analytical Processing (OLAP) is emerging as the most important approach in Data Warehousing. OLAP allows modelling data in a multidimensional way as a cube and to query and analyze data from many different perspectives. Multidimensional OLAP data is presented as a multidimensional data cube. The Multidimensional data cube can be modelled by using facts and set of dimensions based on the notion of dimension hierarchy. In this approach multidimensional cube class is organized into class-hierarchy, which consists of three layers and each layer has an independent function. This approach provides advantages from the several positive aspects of object-oriented modelling, such as security, reusable software, etc. Multidimensional database and OLAP scenarios are becoming very popular these days. OLAP systems impose different requirements than On-line Transactional Processing (OLTP) systems, and therefore, different data models and implementation methods are required for each type of system. There have been several different multidimensional data models proposed recently. However, there are certain key issues in multidimensional modelling, such as derived measures, derived dimension attributes and the additively on fact attributes along dimensions, which are not considered by these proposals.

Keywords: OLAP, Multidimensional Data, Dimensions, Facts*, Data Cube, OOMD*

I. INTRODUCTION

Data Warehouse is frequently organised as collection of multidimensional data cubes, which represent data in the form of data values, called measures, associated with multiple dimensions and their multiple levels. However, some application areas need more expressive model for description of its data. The extension of classical multidimensional model to make Data warehouse more flexible, natural and simple. The concepts and the basic idea was taken from the classical multidimensional model to propose an approach based on Object Oriented Paradigm. In this research Object oriented Data Model is used for description of Data Warehouse data and basic operations over the classical model are provided.

The multidimensional data model is an integral part of On-Line Analytical Processing, or OLAP. Because OLAP is on-line, it must provide answers quickly; analysts pose iterative queries during interactive sessions, not in batch jobs that run overnight. And because OLAP is also analytic, the queries are complex. The multidimensional data model is designed to solve complex queries in real time. The multidimensional data model is important because it enforces simplicity. The central attraction of the dimensional model of a business is its simplicity that simplicity is the fundamental key that allows users to understand databases, and allows software to navigate databases efficiently. The multidimensional data model is composed of logical cubes, measures, dimensions, hierarchies, levels, and attributes. The simplicity of the model is inherent because it defines objects that represent real-world business entities. Analysts know which business measures they are interested in examining, which dimensions and attributes make the data meaningful, and how the dimensions of their business are organized into levels and hierarchies. On-line Analytical Processing (OLAP) is emerging as the most important approach **in** Data Warehousing. **OLAP** allows modelling data in a multidimensional way as a cube and to query and analyze data from many different perspectives. Independent from the different implementation aspects, OLAP data are presented to the user in a multidimensional data model [2]. There are several ways how to formally define multidimensional models and their query languages. However until now there do not exist a commonly accepted formal multidimensional data model. Such a model is necessary **as** a basis for an accepted standardized logical data model for OLAP data. This would allow practitioners and researchers to specify their data warehouses in a unified way. The aim of this paper is to propose an approach for a meta model of multidimensional data with capability to describe objects and the OLAP operations based on the object-oriented model. It seems to the authors that such a model would very much correspond with the original intuition of Codd, when he introduced the concept of OLAP in his pioneering white paper [2].

II. OLAP DATA MODEL

A. *Fact and Dimensions*

Let *L* = be a set of level names, where each level *I* **E** *L* is associated with a set of values ***dom(I);*** let *D* = ***{d,;.-,d,}*** be as finite set of dimensions ***(D);***an d let *F* = *{fi,...,f,}* a finite set of fact names ***(F).***

Definition 2.1. A dimension schema is denoted as a triple (Dname,Ld,i), where DnameEDis the name of dimension, Ld lis a finite set of levels, which contains the distinguished level name All, such that dom(A//) = {a//}. 5 is a partial. Order relation defined among the levels of d. A dimension instance is denoted as a pair (D,R-UP),where D is a dimension schema, and R-UPis a set of partial functions satisfying the following conditions: a. For each pair of levels /1,/2 such that /ls/2 , the roll-up function maps each element of dom(ll) to an element of denoted by R-UP;' :dom(i,)+dom(/2) Given level Ll,/', and l2 such that i 1 i /'5 /2 , the function R - UPfL equals the composition I , R -UP) 0 R - UP('. This implies that: b. For each level /, the function R-UP; is the identity on dam(/); If a level /1 rolls up to in different ways, then the elements of i1 roll up to elements of /,in a consistent way.

III. OLAP MULTIDIMENSIONAL DATA MODEL BASED ON THE OBJECT-ORIENTED APPROACH

A. *The Basic Idea*

To model multidimensional data in an object-oriented way we define the ***Fact*** layer, the ***Dimension*** layer, and the ***Cube*** layer to distinguish its independent function. The ***Fact*** layer is defined as a collection of classes that reflect the dynamic aspect of the multidimensional data and to serve as a basis of calculation. The ***Dimension*** layer **is** defined as a collection of classes for providing the characteristics of the factual data and for providing methods (i.e., rolling-up and drilling-down operation). The ***Cube*** layer built from the ***Fact*** layer and the ***Dimension*** layer is defined as a class **to** allow a subsequent data analysis and to provide the cube operator method. Link between ***Fact*** layer and ***Dimension*** layer is provided by the association relationship**.**

B. *Definition of Model*

[a] Basic Concepts

Let ***A*** = *{a1;..,an)* be a set of attributes names of the class-hierarchy, where each attribute *a* **E** ***A*** reflects an object feature of a specific class and is associated with a set of values *dom(a),* called the domain of ***a.*** The set of methods of the class-hierarchies is given by *M* = ***{ml,...,m***, ***n]***w here each ***m*** **E** *M* provides the dynamic property of an object of a specific class. The set of aggregation function of the class-hierarchies is given byF= ***~l,..-,Jn***w***}***h**,**e re each *f* E F is a specific aggregation function of an object (i.e., **SUM).**

[b] The Dimension Layer

[i] The Class Hierarchy

Definition 3.1. Let Ard = {Ard,,...,Ard ,,,ALL} be a set of attributes roll-up and drill-downs, and a partial order relation is defined as 5 , such that Ardr5A rd, 5 ...“ALtLhe,n the class Hierarchy (H) is defined as; a tuple (Ah, Ard,Mh) , where: Ahis an attribute code of the lowest level of hierarchy, where Ah E A . Ard is a set of attributes of the class H, which consists of hierarchy levels, including level name ALL, where Ard c A. Mh is a set of methods that provides the basic operation for an object H (i.e., “new”, “delete”, “set” operations), where Mh E M.

C. *The Class Dimension*

Definition 3.2. Let H = {H,,...,H,) be a set of hierarchy classes, the class Dimension (D) is defined as a tuple (H, Ad, Md6,iMd), where: H is a set of the classes Hierarchy. Ad is an attribute of the ,class D to provide the name of a dimension, where Ad E A. Mdbis a set of operations allowed on the objects of the class D (i.e., new, delete operations), where Mdl, E M Mdis a set of operations of the class D, where Md c M . Md includes the rolling-up (R-UP) and the drilling-down (D-DOWN) operators.

D. *The Fact Layer*

This layer comprises the class Fact (F) and the class Fact Collection (FC). The relationship between the classvF and the class FC is provided by aggregation relationship. Each object of the class FC consists of 1:M objects of the class F.

E. *The Class Fact*

Definition 3.3. Let *A f* = {.fi;..,.f} be a set of the base attributes of the fact that consists of dimensions code, and let *AH = {Ahl,...,Ah,}* be a set of attribute codes of dimension, there exist domain mapping as follows. ***Am*** = *Value Mf = (new, delete).* An instance of the class *Fact* is given.

F. *The Class Fact Collection*

Definition 3.4. Let F={F,;..,F,}be a set of *fact* classes, the class *Fact Collection (FC)* is defined as a tuple *(F, A$, Mfc, Mag),* where: F is a set of the classes *Fact. Afc* is an attributes of the class *FC* to provide the name of a fact, where *Afc* E *A. M j* is a set of basic operations of the class *FC, dom(Ahl)x ...* ***x*** *dom(Ah,)+* ***dom(afi)x*** *...* ***x*** *dom(af,),* ***m,n*** **E** N. We define the class *Fact* ***(F)*** as a tuple *(A f ,Am, M f)* , where where *Mug* c *M.* where *Mfc* E *M* (i.e., new. delete operations).

IV. MULTIDIMENSIONAL MODELLING PROPERTIES

A data cube [4], [8], [6] is constructed from a subset of attributes in the database. Certain attributes are chosen to be measure attributes, i.e., the attributes whose values are of interest. Other attributes are selected as dimensions or functional attributes. The measure attributes are aggregated according to the dimensions The Figure 1 below depicts a practical data cube example; consider a hypothetical database of number of patients and number of diagnoses. This particular data cube has three feature attributes - place, national cancer register, and time - and a single measure attribute – number of localization. On the base of measure attribute may ask other questions. Such as: the most frequently occurrence of localization (using max function), the greatest age group of patients at risk for year or 5 years an example. By selecting cells, planes, or subcubes from the base cuboid, we can analyze count of patients. In total, a d-dimensional base cube is associated with 2d cuboids. Each cuboid represents a unique view of the data at a given level of granularity. Not all these cuboids need actually be present, however, since any cuboid can be computed by aggregating across one or more dimensions in the base cuboid.

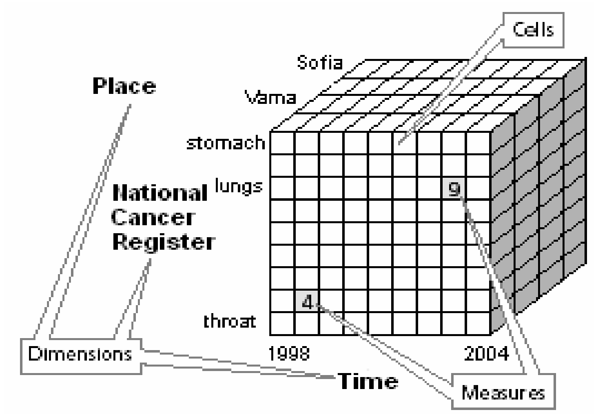

Figure 1. A multidimensional model data cube:- The cube is composed of cells that define fact attributes

A measure is additive along a dimension if they can use the SUM operator to aggregate attribute values along all hierarchies defined on that dimension. The aggregation of some fact attributes—called roll-up in OLAP terminology—might not, however, be semantically meaningful for all measures along all dimensions. Defining the classification hierarchies of certain dimension attributes is crucial because these classification hierarchies provide the basis for the subsequent data analysis. Because a dimension attribute can also be aggregated to more than one other attribute, multiple classification hierarchies and alternative path hierarchies are also relevant. For this reason, directed acyclic graphs provide a common way of representing and analyzing dimensions with their classification hierarchies. Figure 3 shows the different classification hierarchies defined for the *time*, *place, National Cancer Register (diagnoses)* and *Patient* dimensions. On the *National Cancer Register diagnoses)* dimension, a multiple classification hierarchy has been defined so that data values can be aggregated along two different hierarchy paths:

A. National Cancer Register–Diagnoses–Localization–Stage
B. National Cancer Register–Diagnoses–Histology. For the Patient dimension, an alternative path classification hierarchy has been defined with two different paths that converge into the same hierarchy level:
C. Patient–Sex–Branch–Social group
D. Patient–Age group.
   Finally, another alternative path classification hierarchy have been also defined with the following paths for the time dimension:
E. Time–quarter–year–5 years
F. Time–season.
   On the forth fact dimension – Place, classification hierarchy without alternative path have been defined:
G. Place–Town/Village–Municipality–Region–Country.

Once developers define the multidimensional model structure, users can define a set of initial requirements as a starting point for the subsequent data-analysis phase. From these initial requirements, users can apply a set of OLAP operations to the multidimensional view of data for further data analysis. These OLAP operations usually include the following:

[a] ***roll-up,*** which increases the level of aggregation along one or more classification hierarchies;

[b] ***drill-down,*** which decreases the level of aggregation along one or more classification hierarchies;

[c] ***slice-dice,*** which selects and projects the data;

[d] ***Pivoting,*** which reorients the multidimensional data view to allow exchanging dimensions for facts symmetrically.

In this section we summarize the basic concepts of the GOLD model presented in [6] extended in [7]) and provide a brief outline of the latest achievements of previous versions. A more detailed summary definition of the model can be found in the appendix at the end of the paper. The GOLD model is based on the OO paradigm and a MDB schema is defined by *dimension classes* (DC), *fact classes* (FC), *cube classes* (CC) and *views* (V). DC contains *dimension objects* (DO) that provide characteristics of the factual data, while FC contains *fact objects* (FO) that represent the factual data itself. *Fact classes* are specified as *composite classes* in an aggregation relation where *dimension classes* are the components. *Cube classes,* to which OLAP operations are applied to allow us to accomplish a subsequent data analysis, are then defined from these DC and FC, based on user requirements. As we have already mentioned in the introduction, there is no multidimensional data model that considers derived measures and derived dimension attributes, since they only consider static pro entries of data, and therefore, data functionality and behaviour cannot be considered. With our model, however, both data functionality and behaviour can be considered, as it uses the OO paradigm. The GOLD model, therefore, defines complex predicates to build derived attributes, both in dimensions and in facts, by applying both arithmetic operations and relational grouping functions. On the other hand, apart from the conceptual model presented in [11], none of the multidimensional data models presented so far consider the additivity on fact attributes along dimensions.

The GOLD model, however, introduces a definition called Aggregation Patterns (AP) on fact attributes to represent this additivity. Fact attributes can therefore be additive if aggregation operations can be applied along all dimensions, semi-additive if it is not additive along all dimensions, and non-additive if it is not additive along any of the dimensions. With reference to the attribute classification hierarchy, the GOLD model defines dimension attributes as a directed, acyclic and weakly connected graph in which each edge represents a to-one relationship between attributes. We can therefore distinguish between Attribute Roll-up Relation Paths (ARRP) and Attribute Classification Paths (ACP), depending on whether there is a classification hierarchy defined on dimension attributes along the path or not.

For another thing, users can query the database basic schema formed by *dimension classes* and *fact classes*. We define a *cube class* for each basic requirement that the user wishes to execute on the database basic schema. These *cube classes* will encapsulate not only data but also operations allowed on objects in the given class. Thus, OLAP operations (*roll-up, drill-down, slice-dice* and *pivoting*) that allow us to accomplish a subsequent data analysis are considered as public methods of *cube classes*. By applying these OLAP operations, new *views* are defined on *cube classes* that share the same basic properties, i.e. OLAP operations can also be applied to these *views*. In [7] we defined *roll-up, drill-down* and *slice-dice* operations to be applied each time on one dimension of the *cube class*. As previous concepts, we denote A as a set of attributes (a1, a2, ..., an). This set of attributes is defined on domains or primitive classes. Domains consist of a set of values and a set of operations allowed on these values. The domain type is the set of values of the corresponding Data Abstract Type (DAT). Instances of these domains are always in the system, they are neither created nor destroyed and they never change. As basic examples we have the well-known integer and Boolean with their trivial types {0, 1, 2, ...} and {true, false} respectively. We now present the definition of the *product dimension class* for the example in section 2, as an example of the GOLD model. We also represent all paths within the graph to clarify the distinction between ARRP and CP1.

V. CONCLUSIONS

The research in object-oriented data warehousing field is quite new and promising. Many important issues still remain unexplored and deserve further investigation. Initially, concept of object-oriented data warehousing was proposed. Here again two object oriented data-warehousing models and their view maintenance algorithms for efficient ***OLAP*** both in single data-source and multiple data-source environments have been proposed. In this paper, we have proposed the composite data model to create the classes and instances in the warehouse. Since view created and deletion of the data warehouse is very important to the on-line analysis processing, two algorithms have also been proposed to process them. The proposed data model *can* improve the query performance since a new class is formed for each view. The model *can* also keep the original inheritance structure by copying it from the source database. The above proposed method is thus a feasible solution to the object-oriented data warehouse. In the future, we will continue our research on this field and try to build an experimental environment for further investigations. We will also try to propose other flexible data models to fit different application domains. In this paper we have presented the GOLD model, an Object Oriented multidimensional data model, and have demonstrated that the GOLD model can achieve better standards than the models proposed up to now. Examples of these achievements are the consideration of derived measures, derived dimension attributes, the additivity on fact attributes along dimensions and the encapsulation of data with its operations in classes. We are now convinced that the application of the OO paradigm allows us to consider all the key issues in multidimensional modelling that other approach hardly consider, as in case of previous studies presented cannot consider either data functionality or behaviour.

VI. REFERENCES

[1] Bertino E.: Method precomputation in object–oriented databases. Proceedings of ACM– SIGOIS and IEEE–TC–OA International Conference on Organizational Computing Systems, 1991.

[2] Dobrovnik M., Eder J.: Logical data independence and modularity through views in OODBMS. Proceedings of the Engineering Systems Design and Analysis Conference, Vol. 2, 1996, pp. 13-20

[3] Eder J., Frank H., Liebhart W.: Optimization of Object–Oriented Queries by Inverse Methods. Proceedings of East/West Database Workshop, Austria, 1994

[4] Bertino E., Catania B., Chiesa L.: Definition and Analysis of Index Organizations for Object-Oriented Database Systems. Information Systems, Vol.23, No.2, pp. 65-108, 1998

[5] Eder J., Frank H., Liebhart W.: Optimization of Object–Oriented Queries by Inverse Methods. Proceedings of East/West Database Workshop, Austria, 1994

[6] Second East-European Conference on Advances in Databases and Information Systems – ADBIS'98. Poland, 1998, LNCS No. 1475, pp. 260-271

[7] Gray W. Hansen. James V. Hansen, “Data Base Management System”.

[8] Gupta A., Mumick I.S. (eds.): Materialized Views: Techniques, Implementations, and Applications. The MIT Press, 1999.

[9] Terry Montlick, “Object Oriented Software”, Copyright 1995-99 by software Design Consultants, LLC.

[10] E. Bertino et al, Object Oriented Query Languages: The Notion and Issues, IEEE TKDE, vol4, No.3, June 1992.