Metadata for Object-Relational Data Warehouse

Thanh N. Huynh, Oscar Mangisengi, A Min Tjoa
Institute of Software Technology (E188)
Vienna University of Technology
Favoritenstrasse 9-11/188, A-1040 Vienna, Austria
({thanh, oscar, tjoa}@ifs.tuwien.ac.at)

Abstract

For developing data warehouse (DW) and On-Line Analytical Processing (OLAP) systems, the dominant relational database reaches its limitations. On the way of the development, object-relational (O-R) database is preferred to get over those ones. This paper introduces metadata for data warehouse system on O-R database and specifies new kind of metadata for mapping from object-oriented environment to relational environment. We also present the storage structure for repository this new kind of metadata in O-R database.

Keywords: Metadata, OLAP, Data warehouse, Object-relational database.

1 Introduction

The data stored in DW and OLAP systems is collected, integrated and centralized from various operational data store systems. For analysis purpose of the enterprise, the data are usually stored in multidimensional structures [TrPa98], [ReBS97], [Kimb98]. These structures are suitable for analysis purposes since they represent in an intuitive way the factual data according to the characteristics that are considered relevant to the analysis.

For developing the DW and OLAP systems, the dominant relational database reaches its limitations [GoLK99]. On the way of the development, O-R database [Ston95], [Ston97], [OHUS96], [KrBN99] is preferred to get over those ones.

In these systems, metadata plays an important role and provides the foundation for all actions in all stages. It can be considered as glue sticking together all individual parts of these systems.

In this paper, we propose our O-R data warehouse architecture with new metadata layer and describe the design and implementation new kind of metadata to bridge gap between object-oriented environment and relational database.

The paper is constructed as follow. Section 2 discusses the related works, which cover an overview on data warehouse modeling and metadata for data warehouse. The next section shortly reviews O-R databases and its query. An O-R data warehouse is presented in section 4. Section 5 discusses the metadata for the O-R DW. The last section comes with the conclusion.

2 Related Works

There has been a substantial amount of work on the general topic of data warehouse and OLAP. For the sake of relevance and brevity, we discuss generally here only the works that propose metadata for the data warehouse and data warehouse modeling.

Orr in [Orr96] introduces data warehouse architecture with 8 layers including a metadata layer. These layers represent the overall structure of data, communication, processing and presentation that exists for end user computing within the enterprise. Gupta proposed opinion that "the data warehouse model needs to be extensible and structured such that the data from different applications can be added as a business" [Gupt97]. Different approaches to develop a data warehouse were suggested in [Fire97b]. These approaches show us various data warehouse models. Furthermore, Wu and Buchmann proposed logical and physical data warehouse architectures in [WuBu97]. The logical architecture is independent from application and front-end tools. The physical architectures are a mapping

Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000)
Stockholm, Sweden, June 5-6, 2000
(M. Jeusfeld, H. Shu, M. Staudt, G. Vossen, eds.)
http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-28/

of the logical architecture to multidimensional database management system (MDBMS) and relational DBMS (RDBMS).

Kimball et al. proposed data warehouses with a "bus architecture" based on "conformed dimension" and "standard fact" definitions. This is a practical, flexible architecture for data warehouse systems. Furthermore, they proposed a centralized metadata using for the both front room and back room [KRRT98].

Architecture for distributed OLAP is also investigated in ongoing CubeStar project. In this project, also dynamic metadata is distributed in the system [AlGL98].

Different extended relational concepts to model metadata for data warehousing are introduced in [MaTW99]. The differences of the models show a huge advantage of the extended relational model.

3 Object-Relational Database

Nowadays, leading DBMS vendors have committed to O-R DBMS, e.g., Oracle with Oracle8i or Informix issuing Informix-Universal server. Nearly all of them support the Java programming language, which provides an object environment to users of their system.

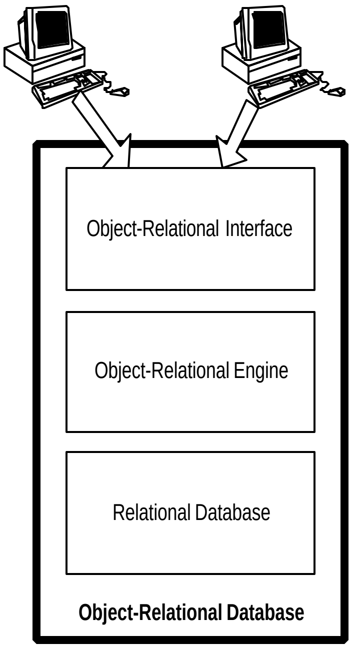

Figure 1: Object-Relational Database

In general, we can regard O-RDBMS architecture as shown in figure 1. Like any other systems, O-R interfaces obtain requirement data and deliver the corresponding object data from the O-RDBMS to the applications. These interface components ensure a transparent access from application outside the system to data storage in its databases.

The Object-Relational engine is object-based environment and bridges the object environment and relational database. It not only manages the native SQL data types (such as integer, number, date, char) but also object data types, which are user-defined or system-predefined object types. Like any classes in an object-oriented programming language, these object types include 'attributes' holding the data and 'methods' manipulating their behaviors. Consequently, the object-relational query language trends to support user-defined functions and operators. Up to now, the SQL3 [Kulk94], [FDCM+99] is preferred to become a standard for object-relational query language but it is still not powerful enough to play this role.

For example, given the object-relational schema and a typical object-relational query [Ston97]:

Create EMP-OR (name=C12, age=int, salary=int, dept=C12, location=point, picture=image);

```
Select name
Form EMP-OR
Where beard (picture) > 0.7 and
        Age > 60 and
        Location in circle ("10,10", 5);
```

Comparing with traditional relation, the two new additional fields that hold data in two new data types are "geographic point" and "image". In the query, "*beard(picture)*" and "*in*" are user-defined operators.

4 O-R Data Warehouse

In this section we propose O-R DW architecture, given in figure 2, based on logical architecture proposed in [WuBu97]. The differences of these architectures are "the object-orientation" approach and the new metadata layer.

With the object-oriented approach, most layers of this architecture -but the "Data Store" layer- consist of many objects of various object types, which perform underlying functions of each component.

In this architecture, the data flow is similar to other data warehouse architectures [KRRT98], [Fire97a], [Orr96], [WuBu97] where data is collected from diverse operational database systems, summarized, aggregated and integrated in a data warehouse, and used as read-only data to supports complex analysis.

This architecture consists of the components described as follows:

1. *Application interface layer*:

In the application interface layer, the objects of this component hide complex data processes from the data warehouse users. The objects of this component are classified into various groups serving different services. Based on their functionalities, each service responds to corresponding requests of third party applications, data analyzers, or other users of the data warehouse system.

For the usage of the data warehouse, the main functions of the object types of this layer are to receive users' queries, preprocess these queries and then send final request to the Data Warehouse Management component. Afterward, they obtain the queries results from the deeper layer. In this architecture, a query is not directly executed at this layer.

For administrating the operations of the data warehouse, the objects of this layer will provide functions to manage user services, control the updating, maintaining processes of the data warehouse. That means, new user services can be added in this layer to support new user requirements if needed.

2. *Data Acquisition:*

The Data Acquisition component can be considered as a tool that constructs the data engine of the data warehouse. The data acquisition objects will extract, transform and transfer data from different legacy operational data stores (ODS) to the data warehouse O-R database.

The functions of this component are divided into suitable sub-function levels that are performed by pattern object types, e.g., this component has various classes, such as: ExtractingService, Transforming-Service, LoadingService, etc.

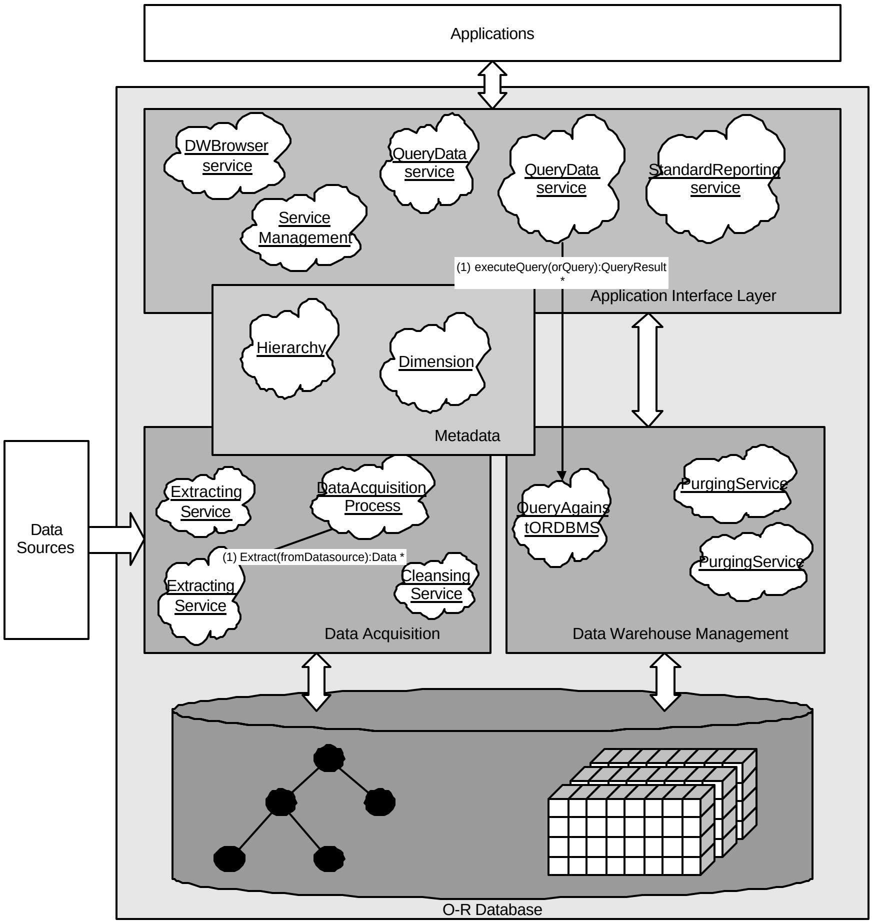

Figure 2: The O-RDW physical Architecture

3. *Data Warehouse Management:*

As a component of the data management layer, this component directly accesses data of the data warehouse from the O-R database. It provides services, which bridge the application interface layer and the O-R database.

In this component, different methods can be applied to access data stored in the O-R database. Furthermore, the database access methods can be updated or added to improve the performance of the data warehouse.

The division of the data management layer into two individual components allows us to clearly distinguish between read-only data processing in data warehouse and data input processing. The functions of this component are mainly to read available data, and to create new materialized views based on this data.

4. *Metadata:*

With regard to metadata in an object-oriented way, we define the behaviors for metadata objects depended on its roles. For instance, metadata can itself count its accessed frequency, make statistics of query usages, and so on. That means that many questions about the warehouse operations can be easily answered by directly querying metadata, e.g., how many reports were created in a day? How often is one kind of data used?

This metadata layer will be discussed in more detail later, in section 5, "O-R Data Warehouse Metadata".

5. Data store:

The data stored in O-R DW differs primarily from DW in relational environment and object-oriented data warehouse. Depending on the requirements and data types, O-R DW designers can decide to model it as a "cube", like MOLAP (Multidimensional OLAP), or as object hierarchy, like O3LAP (Object-Oriented OLAP). For instance, in O-R DW, simple data can be modeled in multidimensional structures looking like what have done in relational database systems [KRRT98], [WuBu97], [Fire97b]. Otherwise, complex data, user-define data can be modeled in object hierarchical structures as suggested for OODBMS [BeMa93]. Furthermore, the objects of any layers, particularly metadata objects, can be modeled in the O-R database.

5 O-R Data Warehouse Metadata

"Metadata is data about data", this definition is too general to give someone the concept of metadata in a data warehouse system. In section 5.1, we summarize some metadata classification in the data warehouse system, proposing new kinds of metadata that exist only in O-R environment. A short description in section 5.2 discusses the multidimensional star schema in relational database. The star schema is used as a frame describing new kind of metadata in the next two sections, 5.3 and 5.4, which give the realization way to design and implement the new kind of metadata, store their attributes in O-R data warehouse.

5.1 Metadata Classifications

There are many kinds of metadata in a data warehouse system [KRRT98], [Kimb98]. Instead of listing them, we prefer to generally summarize existing metadata classifications in various points of views.

In [CoBA99], metadata is classified based on the datawarehouse architecture layers as follow:

- Metadata associated with data loading and transformation. It describes the source data and any changes that were made to the data.
- Metadata associated with data management. It defines the data store in the data warehouse. Every object in the database needs to be described including the data in each table, index, and view, and any associated constraints. This information is held in the DBMS system catalog; however, there are additional requirements for the purposes of the warehouse.
- Metadata used by the query manager to generate an appropriate query. The query manager generates additional metadata about the queries that are run, which can be used to generate a history on all the queries and a query profile for each user, group of users, or the data warehouse.

The other classification divides metadata into technical metadata, business metadata and information navigator metadata [Que97]:

- Technical metadata primarily supports technical staff that must implement and deploy the data warehouse. The information contained within the technical directory is compatible with this kind of audience and contains the term and definition of metadata, exactly as they appear in operational databases.
- The business metadata primarily supports business end users who do not have a technical background, and cannot use the technical metadata to determine what information is stored inside the data warehouse.
- The information navigator metadata is a facility that allows users to browse through both the business metadata and the data inside the data warehouse.

Moreover, the metadata can be considered as two classes, namely static and dynamic.

- Static metadata: This kind of metadata is used to document or browse in this system. E.g., metadata of a dimension. The content of this metadata is fixed in the data warehouse.
- Dynamic metadata: vice versa to static metadata, dynamic metadata is metadata that can be generated and maintained in run time. For instance, metadata of a new frequent access query.

Similarly to any data warehouse in relational, multidimensional or object-oriented databases, the O-R data warehouse also has these kinds of metadata. Referring to the O-R database section, the Object-Relational Engine is object-based environment; in the meanwhile, the data is stored in relational database. Therefore, a new kind of metadata that takes care of the mapping between object environment and relational database must be held in this system.

5.2 A Star Schema in Relational Database

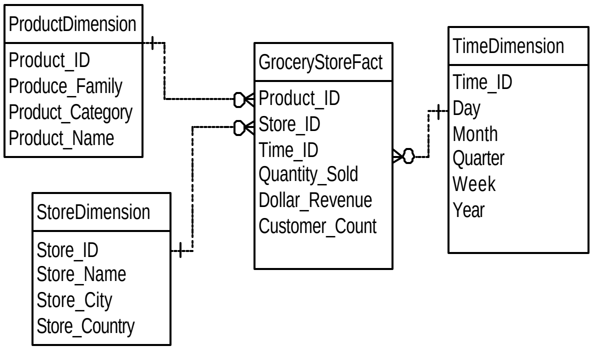

Figure 3: An example of the star schema

The name "dimensional modeling" is considered as a way to make database simple and understandable, particularly for business information analysts. This modeling includes fact and dimensions, which usually describe in a star schema. In relational database, every dimension or fact is stored in table. For example, to describe a star schema in figure 3 we have 4 tables: ProductDimension, StoreDimension, TimeDimension and GroceryStoreFact with their corresponding attribute columns. Furthermore, the schema metadata that represents the dimension structures must be stored somewhere in a table.

5.3 Metadata for Star Schema in O-R DW

Beside many kinds of metadata, in O-R environment, we propose a new kind of metadata that maps and bridge gap between object environment and relational environment. In the limit of this paper, we suggest an approach to realize this kind of metadata for O-R DW. Based on the facility with supporting of O-R database vendors to Java programming language, we also code all our examples in this language.

Generally, the Metadata class is the base of any other metadata subclasses. It includes essential attributes and methods of the metadata subclass. Given the Metadata class in Java language as follow:

```
public class Metadata {
// Base attribute and method of metadata
            String Description;
            int AccessTime=0;
            …
private void increaseAccessTime () {…};
public int getAccessTime () {…};
public String getDescription (){…};
….
}
```

In a metadata object, meta-information of this metadata object can be created and maintained in the object itself, e.g., the *AccessTime* attribute in metadata class.

Starting from the atom item of a relation, a column, we define Column class. An object of this class will be on behalf of an attribute in a relation. More details of this class can be found in Column class document in JBuilder software.

```
public class Column extends Metadata {
// Describing a column of a relation
            String columnName;
            String dataType;
            ….
}
```

In a relation table, there is no distinction from the order of an attribute. However, in data warehouse and OLAP systems, the presentation of data in hierarchy is needed for analytical processing. Therefore we need a mechanism to describe the data structure. In our approach, the metadata Hierarchy class realizes this function. It holds a link list of attributes, see figure 4, in a predefined order, which quite depends on the point of view on the structures of a dimension.

```
public class Hierarchy extends Metadata {
 // Describing a hierarchy of a dimension
            // link list of Column object
            Vector ListOfLevels;
public void insertColumnAt(int at, Column col) {…};
….
}
```

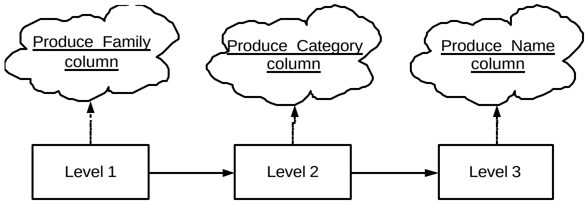

Figure 4:A hierarchy of ProductDimension object

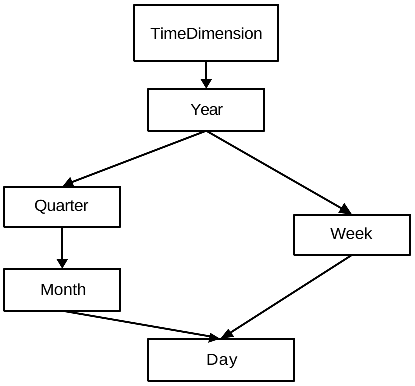

Figure 5: Hierarchies of TimeDimension

With a multi-hierarchy dimension, e.g. multi-hierarchy *TimeDimension* (figure 5), it requires a dimension object of Dimension class holding more than one Hierarchy objects. In Java language, we can realize this requirement by using a link list of objects. Let define the Dimension class as follow:

```
public class Dimension extends Metadata {
 // Describing a dimension
            String dimensionName;
            String dimensionTableName;
            // link list of attributes of the dimension
            Vector dimensionAttributeList;
            // link list of Hierarchy object
            Vector listOfHierarchies ;
            …
public Hierarchy getHierarchyAt(int at) {…};
public Column getAttribute(String attName) {…};
…
}
```

The one-to-one mapping from attributes of the dimension relation to the dimension attribute list is held in dimensionAttributeList attribute, and the listOfHierarchies attribute is used to store list of Hierarchy objects of the dimension.

Now, in turn of FactTable class, it is defined to hold two lists of attributes. They are a dimension list being as a list of Dimension objects and a fact list being as a list of Column objects.

```
public class FactTable extends Metadata {
            String factTableName;
            Vector listOfDimension;
            Vector listOfFact;
            …
}
```

Based on the definition of these classes, a star schema of a fact table can be formed in object schema. Given in figure 6, we have the object schema of the *GroceryStoreFact* fact table. The highest level is *GroceryStoreFact* object, which associates to three dimensions, *TimeDimension*, *StoreDimension* and *ProductDimension*. Each dimension has its own Hierarchy object(s).

Moreover, dynamic metadata for O-R data warehouse can be also created and managed, for instance, to manage some frequent accessed queries. A metadata object mapping the query-to-query result is defined as follow:

```
public class QueryResult extends Metadata {
            String queryString;
            String tableName;
            …
public boolean matchQuery( String qString) {…};
…
}
```

5.4 Metadata Storage in Relational Database

For storage, status of metadata objects are also stored and managed in the relational database. Although, some database vendors support to work with O-O programming languages, e.g., Java, storing codes of object methods in relational database usually require a complex process to load or restore these codes. In our approach, only attributes of these objects are stored. The following tables (from table 1 to table 9) describe the storage repository.

States of Metadata objects are stored in the Metadata table. At defining, all objects of subclasses of Metadata class are Metadata objects, i.e., beside their additional attributes; they also include all attributes as a Metadata object. The values of Metadata object attributes are stored in table 1. In a table stored attributes of a sub-class objects, there is a column, named M_id, used to store id of the corresponding Metadata super-objects of these objects.

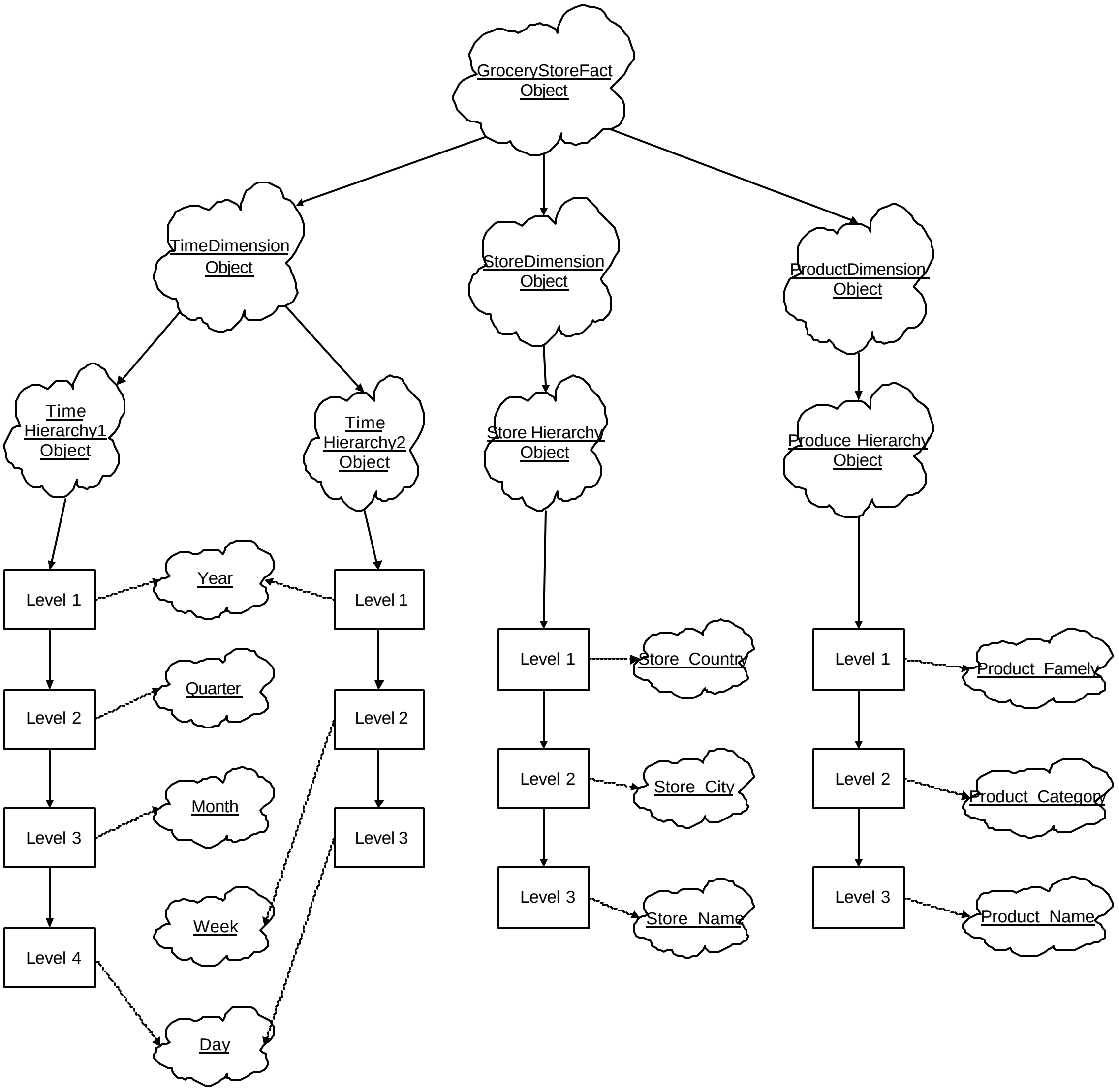

Figure 6: Object schema of the fact table

Table 1: Storage attributes of Metadata objects

M_id	**Description**	**AccessTime**
1	Year	0
2	Quarter	0
3	Month	0
4	Day	0
5	Hierarchy_level1	0
6	Hierarchy_level2	0
7	P_Name	0
8	P_Family	0
…		

Table 2: Storage attributes of Column objects

C_id	**Name**	**Datatype**	**…**	**M_id**
1	Year	Char	…	1
2	Quarter	Number	…	2
3	Month	Char	…	3
4	Day	Number	…	4
5	ProductName	Char	…	7

6	ProductFamily	Char	…	8
7	StoreName	Char	…	17
8	StoreCity	Char	…	18
…				

Table 3 and 4, together hold attributes of all hierarchy objects. The *Hierarchy table* manages to store a link list of attributes ordered in hierarchy objects. The *Hierarchy_Metadata table* presents the relation of *Hierarchy* object with its *Metadata* super class object.

Table 3: Storage of attributes of Hierarchy object

H_id	**C_id**	**H_id_next**
1	1	2
2	2	3
3	3	4
4	4	Null
5	6	6
…		

Table 4: Hierarchy_Metadata table

H_id	**M_id**
1	5
5	14
9	19
…	

The two next tables, 5 and 6, are used to manage the attributes of all dimensions metadata objects.

Table 5: Dimension table

D_id	**Name**	**…**	**M_id**
1	Time	…	9
2	Store	…	10
3	Product	…	15
…			

Table 6: Dimension_Hierarchy table

D_id	**H_id**
1	1
1	9
2	7
3	5
…	

The last three tables, 7, 8 and 9, store the attributes of the *FactTable* metadata objects.

Table 7: FactMetadata table

F_id	**M_id**
1	12
2	13
…	

Table 8: Factlist table

F_id	**C_id**
1	20
1	10
1	22
…	

Table 9: Fact_Dimensionlist table

F_id	**D_id**
1	1
1	2
1	3
…	

6 Conclusions

In this paper, we propose to realize the metadata that shows a mapping between object environment and relational environment in metadata layer of an O-R data warehouse. Various metadata classes are defined and discussed their roles in the O-R data warehouse. The metadata layer and the object-oriented approach together allow us to obtain many powerful characteristics for building an O-R data warehouse.

Comparing to metadata of relational or multidimensional data warehouse systems, this metadata layer plays an active role in maintaining the data in data warehouse. With this mapping metadata, an O-R data warehouse can be really designed and implemented comparing to the object-oriented data warehouse [BuSH98].

References:

[AlGL98] J. Albrecht, H. Guenyel, W. Lehner, An Architecture for Distributed OLAP, International Conference on Parallel and Distributed Processing Techniques and Applications, 1998 (PDPTA'98)

[BeMa93] E. Bertino, L. Martino, Object-Oriented Database Systems: Concepts and Architectures, Addison-Wesley

Publishing Com., 1993.

[BuSH98] J. W. Buzydlowski, Il-Yeol Song, Lewis Hassell, A Framework for Object-Oriented On-Line Analytic Processing, DOLAP 98 workshop, 1998.

[CoBA99] C Connolly, C. Begg, and A. Strachan, Database Systems, A practical Approach to Design, Implementation, and Management, 2nd edition, Addision-Wesley, 1999.

[FDCM+99] Y. Fuh, S. Deßloch, W. Chen, N. M. Mattos, B. T. Tran, et al: Implementation of SQL3 Structured Types with Inheritance and Value Substitutability. VLDB 1999.

[Fire97a] J. M. Firestone, Object-Oriented Data Warehousing, Executive Information Systems, Inc., white paper No. five, August 1997.

[Fire97b] J. M. Firestone, Data warehouses and Data Marts: A Dynamic View, Executive Information Systems, Inc., White paper No. three, March 1997.

[GoLK99] V. Gopalkrishnan, Q. Li, K. Karlapalen, Star/Snow-Flake Schema Driven Object-Relational Data Warehouse Design and Query Processing Strategies, Data Warehousing and Knowledge Discovering First inter. Conf., DaWak'99.

[Grim98] S. Grimes, Modeling Object/ Relational Databases, DBMS, 1998.

[Gupt97] R. Gupta, An Introduction to Data Warehousing, white paper, august 1997, http://www.system-services.com.

[HuTj00] T. N. Huynh, A M. Tjoa, Architecture for Object-Relational Data Warehouse, accepted paper of the ICS2000 in Beijing.

[Kimb98] R. Kimball, Meta Meta Data Data, DBMS magazine, March 1998.

[KrBN99] V. Krishnamurthy, S. Banerjee, A. Nori, Bringing Object-Relational Technology to The Mainstream, SIGMOD Conference 1999: 513-514

[KRRT98] R. Kimball, L. Reeves, W. Thornthwaite, M. Ross, The Data Warehouse Lifecycle Toolkit, John Wiley & Sons, Inc., 1998.

[Kulk94] K. G. Kulkarni: Object-Oriented Extensions in SQL3: A Status Report. SIGMOD Conference 1994: 478

[MaTW99] O. Mangisengi, A M. Tjoa, R. R. Wagner, Metadata for Data Warehouses Using Extended Relational Models Proc. of third IEEE Computer Society Metadata Conference, April 1999.

[OHUS96] M. A. Olson, W. M. Hong, M. Ubell, M. Stonebraker, Query Processing in a Parallel Object-Relational Database System, Data Engineering Bulletin 19(4): 3-10 (1996)

[Orr96] Ken Orr, Data Warehousing Technology, The Ken Orr Institute, A white paper, 1996.

[Que97] Que, The Official Client/Server Computing Guide to Data Warehousing, Que Books, 1997.

[ReBS97] Red Brick Systems Inc., "Star Schema Processing for Complex Queries", white paper 1997.

[Ston95] M. Stonebraker, Object-Relational DBMS- The Next Great Ware, Morgan Kaufman Publishers, San Francisco, CA, 1995.

[Ston97] M. Stonebraker, Architectural Options for Object-Relational DBMSs, Informix White Paper, 1997.

[TrPa98] J. Trujillo, M. Palomar, An Object Oriented Approach to Multi-dimensional Database Conceptual Modeling (OOMD), DOLAP 98 workshop, 1998.

[WuBu97] M. Wu, A. P. Buchmann, Research Issues in Data Warehousing, BTW 1997: 61-82.