Six layers Architecture Model for Object Oriented Database

Mansaf Alam

Department of Computer Science, Jamia Millia Islamia, New Delhi-110025
Delhi, India

Abstract

In this paper, a six layered architecture for object oriented database from access and management point of view is proposed. This architecture model consist of six layers: Interaction layer, Application layer, Administration layer, Security layer, Paging layer, and Virtual layer.

Keywords: Security, layer**,** Architecture Model, Design**,** Paging, Administration.

1. Introduction

The very popular ANSI/SPARC architecture [1] gives three levels that drive the data access and management of a database. These three levels are external, conceptual, and the internal. The external level consists of particular views of data dedicated to particular client applications or particular users. In relational databases the external level implements two kinds of facilities: access privileges to particular resources grated by database administrator to particular users. The SQL views that customize encapsulated and restrict resources to be accessed. This approach has proven to be enough simple and satisfactory for majority of applications of relational databases. The conceptual level is common for the entire database environment. In this level the storage structure of data is defined. The Internal of Physical level describes the physical storage of data on secondary device.

But the situation is different in object oriented database, the complex structure data is managed by object oriented database. Piotr Habela1 et. al. [2] provides foundation for three-level database architecture and correspondingly three database development roles: (1) a database programmer defines stored objects, i.e. their state and behavior; (2) a database administrator (DBA) creates views and interfaces which encapsulate stored objects and possibly limit access rights on them; (3) an application programmer or a user receives access and updating grants from DBA in the form of interfaces to views. They present a concrete solution that they developed as a platform for grid and Web applications.

The problem how to efficiently store, retrieve, analyze and modify the biological data and multimedia data are
becoming an important issue for most biological scientists and computer scientists who is working in the field of multimedia data manipulation. In order to solve this problem, a Domain Specific Object Oriented Data Base Management System (DSOODBMS) is designed to manipulate Protein Data that is biological data, Yanchao Wang et. al.[3]. They have designed special architecture for the protein data in object oriented databases.

Muhammad Ubaid et al.[4] have describe in their paper that the object oriented techniques may include features such as encapsulation, modularity, polymorphism, and inheritance, for implementing this paradigm. They suggested that there should be a good architecture design of classes' schema for the OODBM system that could help to maintain the relationship between the objects and fulfill the core concept of the object oriented. When we are storing the data as object in OODBMS than OODBMS maintain the Object Identity (OID) and the identity of an object has an existence independent of the values of the object attributes, Elisa Bertino et. al. [5].

2. 2. Architecture design issues for OODBMS

The architecture play important role in database system to manage the data. The architecture design decisions concern the rules for data transfer from database server to client and client to database server, vice versa. The error handling is also important issue while transferring data
between user and machine. The machine is consisting with physical devices. The error may

occur due to power fluctuation while storing the data in physical memory. The different kinds of users and is independent of databases, therefore users can request data in Object Oriented format, but data can be stored in multiple formats in Object oriented databases system, Yanchao Wang1 et. al. [6]. They used Object oriented database as their middleware part that reduced interpretation work and interpretation time among different language translation. If one chooses some object-oriented database system (OODBMS) to manage the data, then the data needs to be loaded and stored by that system in its internal data formats (i.e. persistent C++ structures). These data formats will not work with legacy software, Arie Shoshani [7]. Similarly, if the data is stored in traditional table formats, that cannot be readily used by C++ programs. Rick Cattell [8], has described in his paper, if someone want the object-oriented programming language integration of object-oriented DBMSs, they can use object/relational mapping solutions, albeit with some performance and convenience drawbacks.

3. Object oriented database layers architecture model

This section describes the layers architecture in object oriented databases. Data are transfer from database server to client, which passes through six layers. These six layers have different responsibility do as per requirement. The fig 1.1 shows the six layer architecture model for object oriented data model.

3.1 Interaction layer

The interaction layer is first layer of Six Layers Architecture Model for Object Oriented Databases. In this layer, user can interact with the databases. The user can send the data to databases as well as data can be retrieved from database to user. Robert Greene [9] has talking about individual operations within a transaction, and then there are lots of possibilities. To that end, I'm not sure what was Mariott's point about millions of objects needing to be locked and the impact of that in the object based architecture.

3.2 Application layer

The application is the second layer in this model. Robert Greene[10] making that assumption, if by chance an early adopter choose an ODB who's architecture was ill suited for their applications needs, the reasoning lead immediately to the conclusion that no ODB was suited to solve their needs. From this illogical thinking came the permeation of misconceptions regarding the OODB: they are too slow, they don't handle high concurrency, and they don't scale with large data.

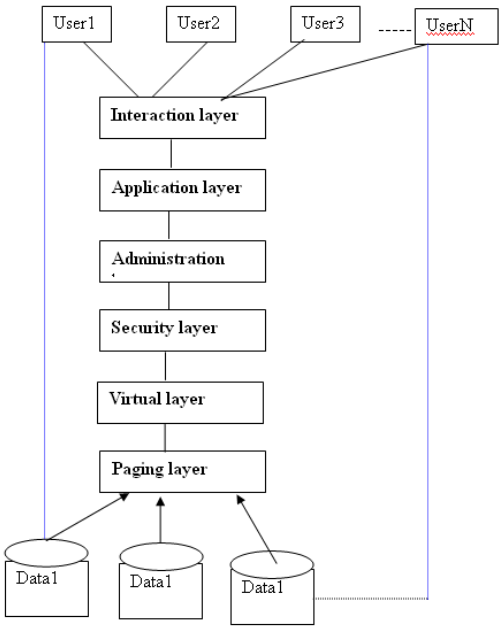

Fig: 1.1 Six layers Architecture Model for Object oriented database

3.3 Administration Layer

This layer is responsible for management of administrative information. This layer can change responsibility as per requirement. Cristina Ribeiroet al.[11] disused the traditional administrative information, pharmaceutical companies, as knowledge intensive organizations, retain a large number of scientific records pertaining to their research and development activities. They also strict requirements of the quality system impose complex data as well as document workflows. The success in preserving complex structures like

databases depends crucially on the amount of information that is lost in the process and this is inversely related to the amount of metadata included in the preservation package. In this

model Administration layer control the flow of data as well as provide the permission to access the data.

3.4 Security Layer

The security layer play important role in this model. The security layer is responsible to provide the full security to data and also provide to the security of application used to manage the data also. David Litchfield[14] examined the differences between the security posture of Microsoft's SQL server and Oracle's RDBMS based upon flaws reported by external security researchers and since fixed by the vendor. This layer can provide the authentication to the uses as well as the authentication to databases administrators. All the security concerned is considered in this layer. Who can use what type of data.

3.5 Virtual Layer

M. Abdelguerfi et. al[12] have shown that the main advantage in their approach is that the memory requirement of each slice processor is very small and is independent of input size. In this model the virtual layer manage the data virtually. This time the large volume of data are managed. The concept of virtual is to put the data outside the memory. As per the requirement the data are converted in real memory. In this ways, the problem to manage large data is solved.

3.6 Paging Layer

M. K. Mohania and N. L. Sarda [13] described three level architecture for a DDedDBS which addresses the problems of partitioning and distributing a large rule base and efficient query handling. The paging layer is responsible to divide the data in the form of pages. The pages are managed easily. The data are divided into pages as the same size of page frame; the page frame is that dividing memory in equal number of partitions. In this way large volume of data can be managed efficiently.

4. Conclusions

This model will be beneficial to manage the large and complex data. The large and complex data are managed efficiently in object oriented database management system. This model is developed by considering the all aspect of data. In this model data can passed through different layer and each layer con perform their duties separately.

References

1. D.C. Tsichritzis, A. Klug (eds.): The ANSI/X3/SPARC DBMS Framework: Report of the Study Group on Data Base Management Systems, Information Systems 3, 1978.
2. Piotr Habela1 et. al., Three-Level Object-Oriented Database Architecture Based on Virtual Updateable Views, European Commission under the 6th FP project e-Gov Bus, IST-4-026727-ST.
3. Yanchao Wang, and Rajshekhar Sunderraman, An Algebra for Protein Structure Data, World Academy of Science, Engineering and Technology 56, PP 363-368, 2009.
4. Muhammad Ubaid, Nauman Atique and Saira Begum, A Pattern for the Effective Use of Object Oriented Databases, 978-1-4244-4609-4/09, IEEE, 2009.
5. Elisa Bertino and Lorenzo Martino, Object-Oriented Database System Concepts and Architectures, University of Genoa Lorenzo Martino—ISBN 0-201-62439-7, Pages – 14 to 16, 1993.
6. Yanchao Wang and Rajshekhar Sunderraman, Database Management System for Protein Structure Data, K. Elleithy (ed.), Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering526–531, Springer Science+Business Media B.V. ,2008.
7. Arie Shoshani, Architectural Issues in the HENP Grand Challenge Project.
8. Rick Cattell, Relational Databases, Object Databases, Key-Value Stores, Document Stores, and Extensible Record Stores: A Comparison, ODBMS.ORG, January, 2010.
9. Robert Greene, OODBMS Architectures Defended, Versant Corporation.
10. Robert Greene ,OODBMS ARCHITECTURES ,An examination of implementations V.P. Product Strategy, Versant Corporation.

11. Cristina Ribeiro - Gabriel David, Database Preservation,DEI Faculdade de Engenharia, Universidade do Porto and INESC-Porto, {mcr,gtd}@fe.up.pt,2007.
12. M. Abdelguerfi and A.K. Sood,A Fine-Grain Architecture for Relational Databases Aggregation Operations,IEEE,1991.
13. M. K. Mohania and N. L. Sarda, An Architecture for a Distributed Deductive Database System, IEEE TENCON '93 / Beijing.
14. David Litchfield, Which database is more secure? Oracle vs. Microsoft, An NGSSoftware Insight Security Research (NISR) Publication, Next Generation Security Software Ltd, 2006