# *Architecture In Object Oriented databases.*

**Sunanda Luthra**

Lecturer, Department of CSE/IT, Amritsar College of Engg. & Tech, Amritsar.143001,
*Punjab, India, E-mail :er_sunanda@yahoo.co.in..*

## *Abstract*

This paper reviews the state of the art of Object-Oriented Database Management Systems (OODBMS). The objective of this paper is to provide the reader with an understanding of the issues relevant to OODBMS technology and to describe where commercial products stand on these issues. It further describes the methods to handle the existing database technologies in object-oriented scope.

In today's world, Client-Server applications that rely on a database on the server as a data store while servicing requests from multiple clients are quite commonplace. Most of these applications use a Relational Database Management System (RDBMS) as their data store while using an object oriented programming language for development. This causes a certain inefficiency as objects must be mapped to tuples in the database and vice versa instead of the data being stored in a way that is consistent with the programming model. The "impedance mismatch" caused by having to map objects to tables and vice versa has long been accepted as a necessary performance penalty. This paper is aimed at seeking out an alternative that avoids this penalty.

Keyword and Phrases: OODBMS, Client-Server, RDBMS, *Impedence Mismatch,* objects, mapping of objects to tables.

## 1. The Need for Object-Oriented Databases

The increased emphasis on process integration is a driving force for the adoption of object-oriented database systems. For example:

- The Computer Integrated Manufacturing (CIM) area is focusing heavily on using object-8oriented database technology as the process integration framework.

- Advanced office automation systems use object-oriented database systems to handle hypermedia data.
- Hospital patient care tracking systems use object-oriented database technologies for ease of use.
- Computer Aided Design (CAD), Computer Aided Manufacturing (CAM) and Computer Aided Software Engineering (CASE) applications use object-oriented database to manage the complex graphical data.

All of these applications are characterized by having to manage complex, highly interrelated information, which was difficult to manage in RDBMS. To combat the limitations of RDBMS and meet the challenge of the increasing rise of the internet and the Web, programmers developed object-oriented databases in 1980.

## 2. Object Oriented Usage Areas:

Generally, an object database is a good choice when you have the following three factors: business need, high performance, and complex data.

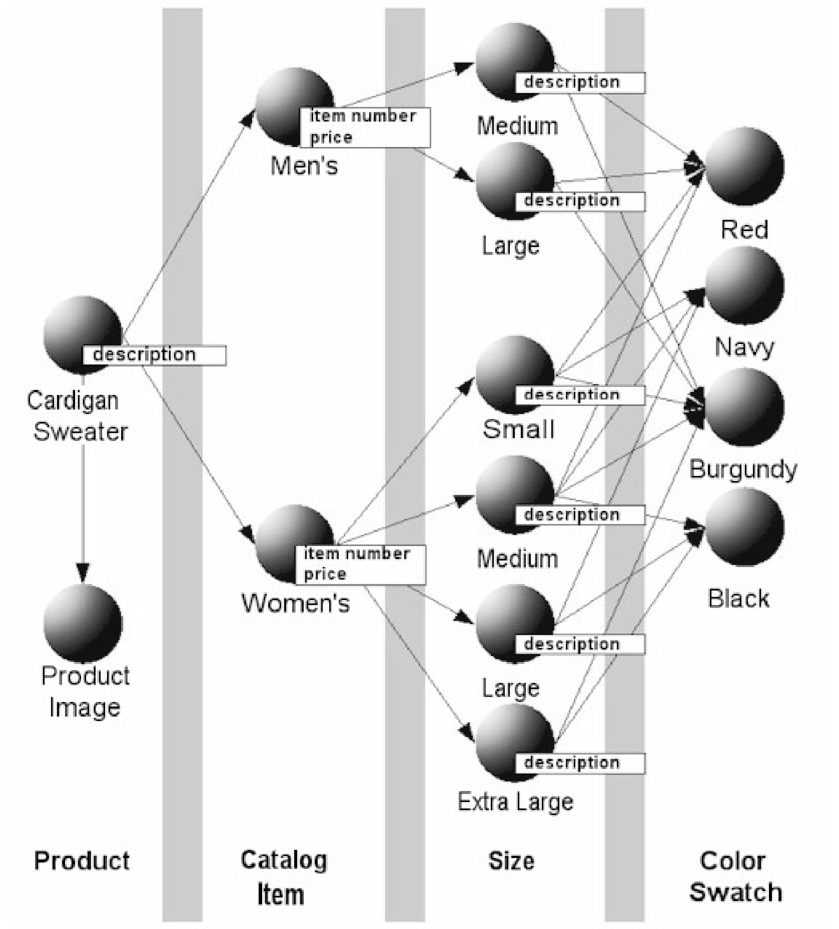
### 2.1 High Performance

With complex data, ODBMS will run anywhere from 10 to 1000 times faster than an RDBMS. The range of this performance advantage depends on the complexity of the data and the access patterns for the data. The reason for high performance is that the Object Oriented Databases are optimized for the traversals related to complex data. They also do not have any "impedance mismatch" when it comes to using object programming languages such as Java and C++.

**2.2 Complex data:** The complex data is often characterized by:

- A lack of unique, natural identification.
- A large number of many-to-many relationships.
- Access using traversals.
- Frequent use of type codes such as those found in the relational schema.

For Example: the following clothing database uses the complex database and objects.



## 3. Overview of Object Oriented Database Management Systems

An OODBMS is the result of combining object oriented programming principles with database management principles. Object oriented programming concepts such as encapsulation, polymorphism and inheritance are enforced as well as database management concepts such as the **ACID Properties** (Atomicity, Consistency, Isolation and Durability) which lead to system integrity, support for an *ad hoc* query language and secondary storage management systems which allow for managing very large amounts of data. Features that are common in the RDBMS world such as transactions, the ability to handle large amounts of data, indexes, deadlock detection, backup and restoration features and data recovery mechanisms also exist in the OODBMS world.

The Object Oriented Database specifically lists the following features as mandatory for a system to support before it can be called an OODBMS:

Complex Objects, Object Identity, Encapsulation, Types and Classes, Class or Type Hierarchies , Overriding, Overloading and Late Binding Computational Completeness, Extensibility, Persistence,Secondary Storage Management , Concurrency, Recovery and an Ad Hoc Query Facility.

A primary feature of an OODBMS is that accessing objects in the database is done in a transparent manner such that interaction with persistent objects is no different from interacting with in-memory objects. This is very different from using RDBMSs in that there is no need to interact via a query sub-language like SQL nor is there a reason to use a Call Level Interface such as ODBC, ADO or JDBC. Database operations typically involve obtaining a database root from the the OODBMS which is usually a data structure like a graph, vector, hash table, or set and traversing it to obtain objects to create, update or delete from the database. When a client requests an object from the database, the object is transferred from the database into the application's cache where it can be used either as a transient value that is disconnected from its representation in the database (updates to the cached object do not affect the object in the database) or it can be used as a mirror of the version in the database in that updates to the object are reflected in the database and changes to object in the database require that the object is refetched from the OODBMS.

## 4. Comparisons of OODBMSs to RDBMSs

***According to Mary Loomis, the architect of the Versant OODBMS:***

- "Relational database design is really a process of trying to figure out how to represent real-world objects within the confines of tables in such a way that good performance results and preserving data integrity is possible.”
- “Object database design is quite different. For the most part, object database design is a fundamental part of the overall application design process. The object classes used by the programming language are the classes used by the ODBMS. Because their

models are consistent, there is no need to transform the program's object model to something unique for the database manager."

There are concepts in the relational database model that are similar to those in the object database model. The equality of the various concepts in RDBMS and OODBMS is as:

| | |
|---|---|
| • Relation or Table | Class. |
| • Tuple | An instance of a class. |
| • Column in a Tuple | Class Attribute. |

## 5. Characteristics of Object-Oriented Databases in Depth

Object-oriented database technology is a marriage of object-oriented programming and database technologies. Figure below illustrates how these programming and database concepts have come together to provide object-oriented databases.

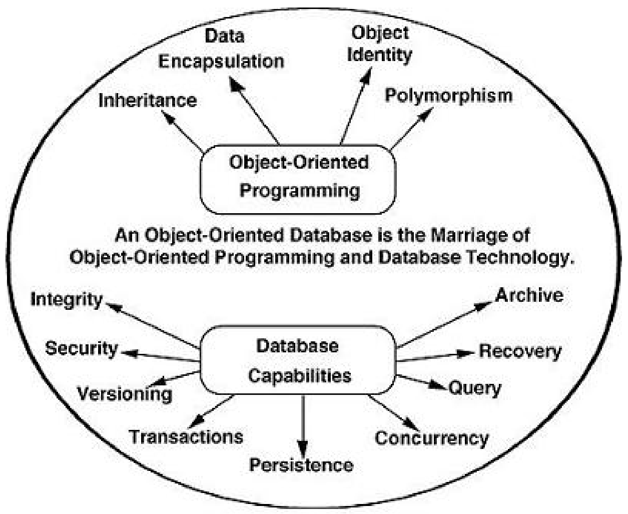


Figure 1. Makeup of an Object-Oriented Database

The most significant characteristic of object-oriented database technology is that it combines object-oriented programming with database technology to provide an integrated application development system. There are many advantages to including the definition of operations with the definition of data:

- The defined operations apply ubiquitously and are not dependent on the particular database application running at the moment.
- The data types can be extended to support complex data such as multi-media by defining new object classes that have operations to support the new kinds of information.

Other strengths of object-oriented modeling are well known. For example, inheritance allows one to develop solutions to complex problems incrementally by defining new objects in terms of previously defined objects.

Polymorphism and dynamic binding allow one to define operations for one object and then to share the specification of the operation with other objects. These objects can further extend this operation to provide behaviors that are unique to those objects. Dynamic binding determines at runtime which of these operations is actually executed, depending on the class of the object requested to perform the operation All of these capabilities come together synergistically to provide significant productivity advantages to database application developers.

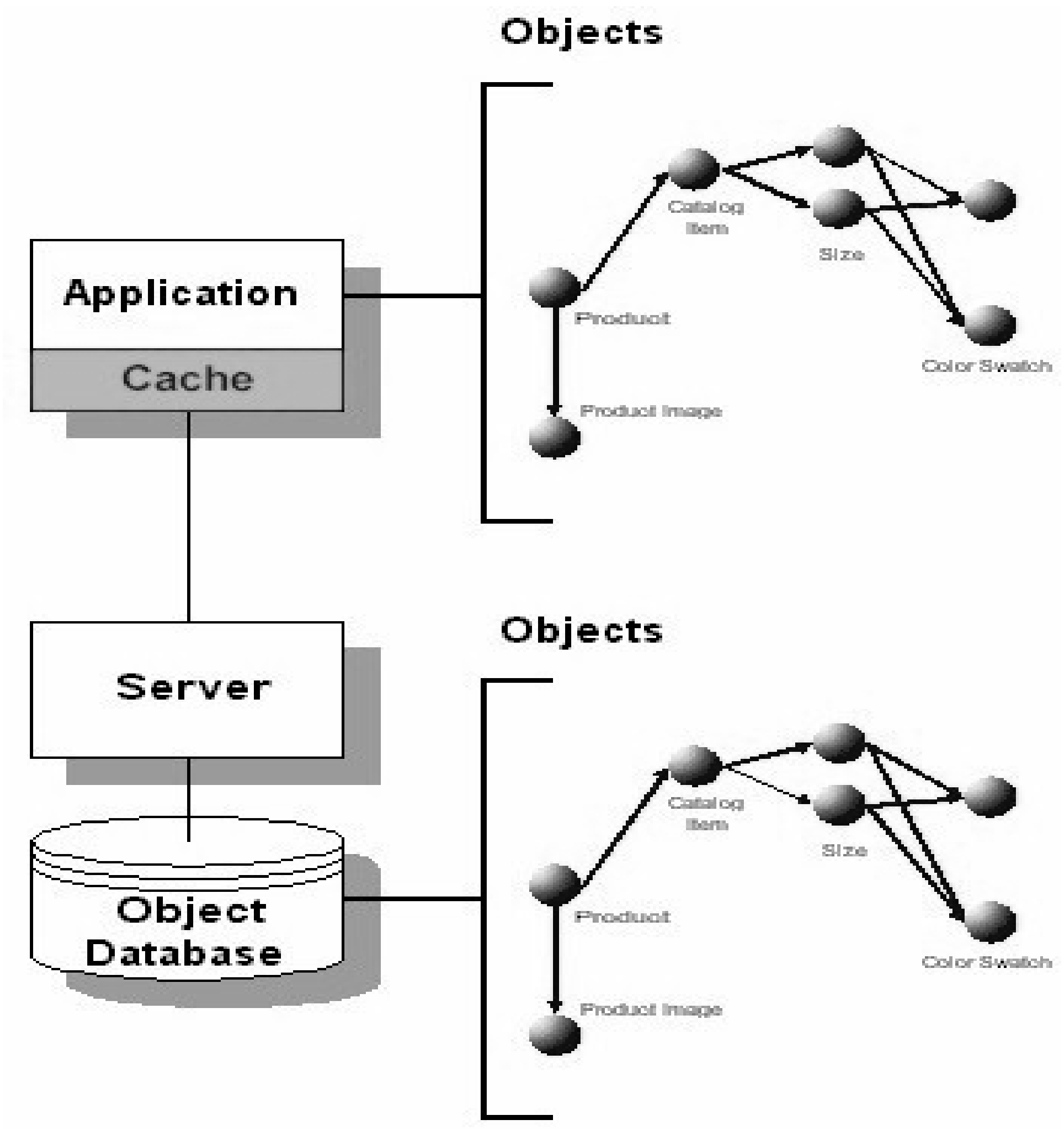
A unique characteristic of objects is that they have an identity that is independent of the state of the object. ***For example,*** if one has a car object and we remodel the car and change its appearance, the engine, the transmission, and the tires so that it looks entirely different, it would still be recognized as the same object we had originally. Within an object-oriented database, one can always ask the question, “is this the same object I had previously?”, assuming one remembers the object’s identity. Object-identity allows objects to be related as well as shared within a distributed computing network.

## 6. Architectures using object Database Products:

### 6.1 Stand-alone architecture

If you are using C++ or Java in a stand-alone application and have the need for a database that provides high performance on complex data, it is difficult to beat an ODBMS. The reason is two-fold:

1. **With an ODBMS, you have only one model to manage,** the model that your object programming language uses. See the diagram below, which shows the same model 2being used in the database and the application. There is also no need to program any mapping between the data in the database and the data in the application.



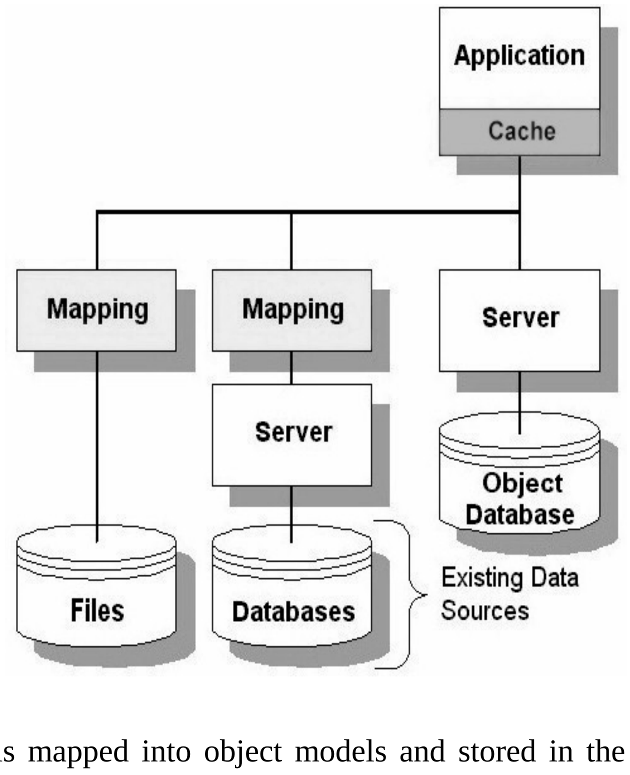
2. **An ODBMS gives you excellent performance on object models**. This means either you can get extreme performance on complex data or you can use less expensive hardware than you might need with a relational DBMS.

Some examples of stand-alone applications:

- Web sites that do not use any existing data.
- Programming Tool.
- Design Tools.
- Multimedia Tools.
- Catalogs on CDs.
- Embedded applications in general.

## 6.2 Architecture with existing data sources

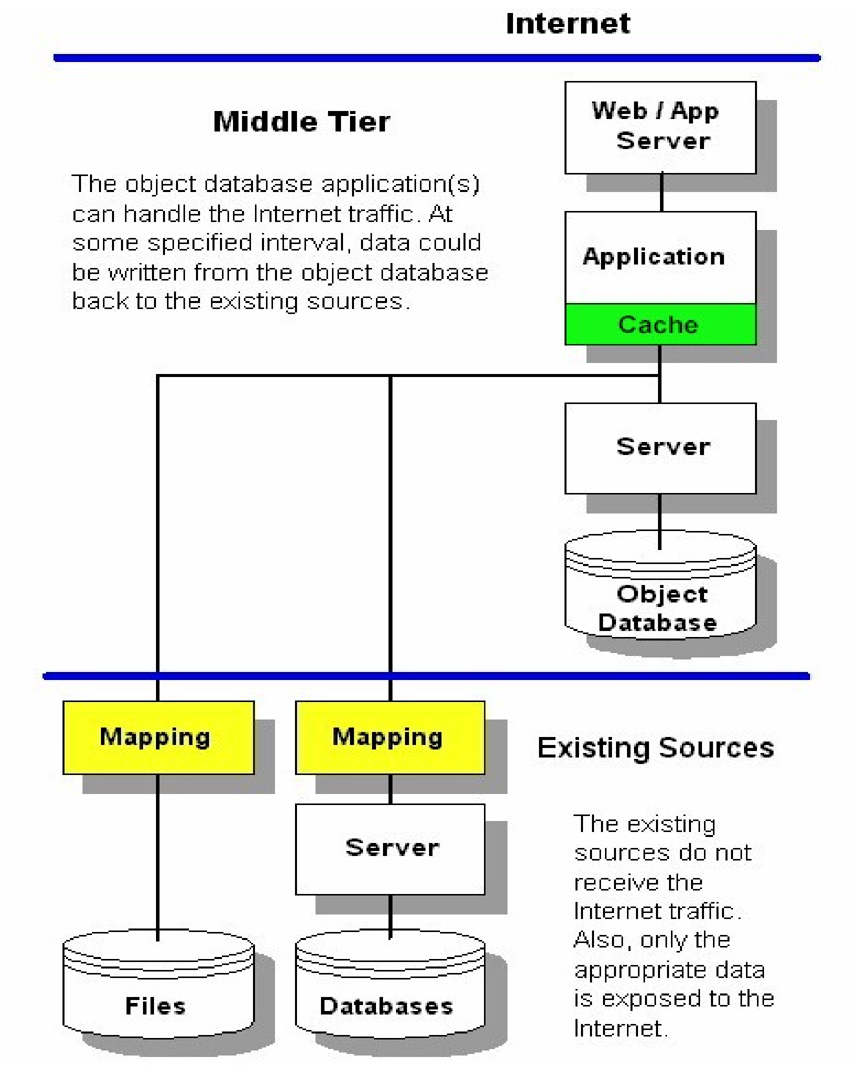
Object databases can be a way of staging data for your C++ or Java applications. This example shows two existing data sources that have data in non-object formats (flat file and relational, for example).



The non-object data is mapped into object models and stored in the object database. This object database now holds some part of the existing data and perhaps some of its own data that did not exist previously. At some later time, the object application can obtain this data and tap the high performance that an object database provides. This performance is a result of having the same model in the object database as is used by the object application.

### 6.2.1 Middle-tier architecture

This middle-tier architecture allows the existing database to be the "database of record." At the same time, it also protects the existing database from direct Internet traffic and provides a high performance engine to interact with the Internet traffic.


Internet
Middle Tier
The object database application(s) can handle the Internet traffic. At some specified interval, data could be written from the object database back to the existing sources.
Web / App Server
Application
Cache
Server
Object Database
Mapping
Mapping
Existing Sources
Server
The existing sources do not receive the Internet traffic. Also, only the appropriate data is exposed to the Internet.
Files
Databases

### 6.2.2 Object-relational mapping

If we want to take advantage of using an ODBMS in the middle tier and have one or more existing relational databases, we will need to map data from a relational format to an object format. This mapping of data can become complex. If we would code this mapping yourself, the amount of code devoted to mapping often becomes 30 to 40 percent of your total code. Unfortunately, that 30 to 40 percent is only mapping, not helping to solving our business problem. It also adds to the possible defects we may need to fix.

So, if we need to map, consider using an object-relational mapping product. They will save our time and reduce the number of defects in our mapping. An object-relational mapping product is shown at the left of figure of middle tier architecture. It handles the mapping and has a cache much like an ODBMS. Data is mapped to the ODBMS and from the ODBMS based on the application needs. The ODBMS provides high-speed performance for the Internet.

## 7. Current Users of ODBMSes:

The following information was gleaned from the ODBMS Facts website.

- The Chicago Stock Exchange manages stock trades via a Versant ODBMS.
- Radio Computing Services is the world's largest radio software company. Its product, Selector, automates the needs of the entire radio station -- from the music library, to the newsroom, to the sales department. RCS uses the POET ODBMS because it enabled RCS to integrate and organize various elements, regardless of data types, in a single program environment.
- The Objectivity/DB ODBMS is used as a data repository for system component naming, satellite mission planning data, and orbital management data deployed by Motorola in The Iridium System.
- The ObjectStore ODBMS is used in SouthWest Airline's Home Gate to provide self service to travelers through the Internet.

- Ajou University Medical Center in South Korea uses InterSystems' Cachè ODBMS to support all hospital functions including mission-critical departments such as pathology, laboratory, blood bank, pharmacy, and X-ray.

- The Large Hadron Collider at CERN in Switzerland uses an Objectivity DB. The database is currently being tested in the hundreds of terabytes at data rates up to 35 MB/second.

- As of November, 2000, the Stanford Linear Accelerator Center (SLAC) stored 169 terabytes of production data using Objectivity/DB. The production data is distributed across several hundred processing nodes and over 30 on-line servers.

## 8. Commercial ODBMSes

- Polyhedra
- Poet
- Gemstone
- Eye-DB
- Objectivity/DB
- Jasmine- CA OO DB
- Itasca
- Metakit- The Structured database which fits into the palm of your hand.
- Conte Xt
- XDb
- ObjectDB

## 9. Freely Available ODBMSes

- SHORE Web Page
- FTP Access to Shore
- Lincks
- Texas Persistent Store Library
- MinneStore(tm) version 2 - "Libre" OO DB for various Smalltalk implementations.

- IronDOC (See also the Old IronDOC Page) provides database-like operators to robustly manage components and roll back changes. The author, <mccusker@best.com>, was apparently involved with the creation of the OpenDoc component scheme, particularly the **Bento** structured storage system. Note that McCusker is now apparently working on an OpenDoc-like system called Bolide.

## 10. Conclusion:

The main objective of this paper is evaluation of the current state of OODBMS products and architecture. This paper was written with two specific purposes. The first was to enumerate and describe the many features that are provided by an object-oriented database. This information should be useful to those readers who are new to object-oriented databases, and even to those new to object oriented technologies and/or database technologies. A second objective was to provide readers a starting point for performing more in-depth OODBMS Architecture evaluations. There is a misunderstanding between the users that shifting from existing database to object-oriented database is a tedious task.. But the fact is that it can be easily shifted to object oriented database using Relational Mapping and Three – Tier Architecture as discussed above.

## References:

1. Data base Concepts By Colloly. By Pearson Education.
2. Database Concepts by Korth by Tata Mcgraw hill
3. WWW.Citeseer.com
4. WWW.odbmsfacts.com
5. WWW.service-architecture.com
6. WWW.wikepaedia.com