THE OBJECT-ORIENTED DATABASES APPROACH REPRESENT A TYPICAL SOLUTION FOR INTEGRATED APPLICATIONS

Yahya Mahdi Hadi Al-mayali
Dept. of Computer\ College of Mathematics and Computer Science\ Kufa University
E-mail: yahyamh@edu.kuiraq.com

ABSTRACTION:

Most database systems initially concentrated on handling typical business data such as: Columns of numbers, short text, and dates. With the advent of powerful and affordable multimedia computers, people increasingly need databases to store more complex objects. Typical objects include graphics, sound, and video clips. For some applications, like geographic Information Systems(GIS) and computer-aided design (CAD), developers need the ability to define their own objects.

Storing objects involves two aspects: storing data for their properties and storing and activating their methods. The first part- storing properties – is relatively easy. In relational database, the properties becomes columns. Relational databases can be adapted to handle objects. On of main components is the ability to define new data types. These data types are treated as objects, and they can inherit properties from other data types.

Many commercial applications are created using OO methodologies. Given that several object-oriented applications might need to share the same data, we describe how such applications can interoperate with each other. By exposing the internal object properties and methods, the developers can use the applications to create the complex integrated business applications. The applications can use all the features of a spreadsheet, word processors, or graphics packages.

This technology also provides function as well as data together so that the data can be manipulated by the given function. we show that how the data is more secure in object-oriented database than in relational database and also why do we migrate from RDBMS into OODBMS.

In this paper, we describe the major characteristics and requirements of object-oriented applications and how they may affect the choice of a system and method for making objects persistent in that application.

Keywords: object-oriented database, Relational database, ODBMS

1. Introduction

Originally, business databases dealt with numbers and short text elements. DBMSs were designed to provide good performance for this type of data. Just look at types of data that can be stored and at the commands available to manipulate that data. Because most business applications were transaction-oriented, this focus worked well. Businesses were content to store simple items such as account balances, product numbers, names, and dates.

In the last few years, computers have been asked to handle increasingly complex types of data, including images, sound, and video. At this pint most of these objects are handled as separate files, with individual programs to create and manipulate this type of data. Because relational databases are so good at handling basic data types, it is natural to ask whether databases can also improve the storage, searching, and manipulation of more complex objects.

The next step involving complex data objects is to determine whether it is possible to create a true object-oriented DBMS. That is, can a DBMS store and manipulate entirely new objects that are defined by the developer? Storing objects is

Another powerful use of objects lies in their ability to create integrated application. Today most business applications are built using a single tool, such as a DBMS, a spreadsheet, or a separate programming language like C+ +. On the other hand, managers use a variety of tools every day, including spreadsheets, word processors, scheduling systems, and e-mail. Increasingly, managers need applications that cross these boundaries and can integrate and share data with each of these tools. For example, a monthly report might pull data from the DBMS, analyze it and create graphs within a spreadsheet, and then format the output using a word processor.

Developers can take advantage of two aspects of this new world:

1. They can create applications that use features from all of these tools, and
2. The objects created with these tools (e.g., a spreadsheet) can be stored within the database.

Remember that databases are designed to share data among many users-features like security and concurrency control can be value valuable for sharing complex objects

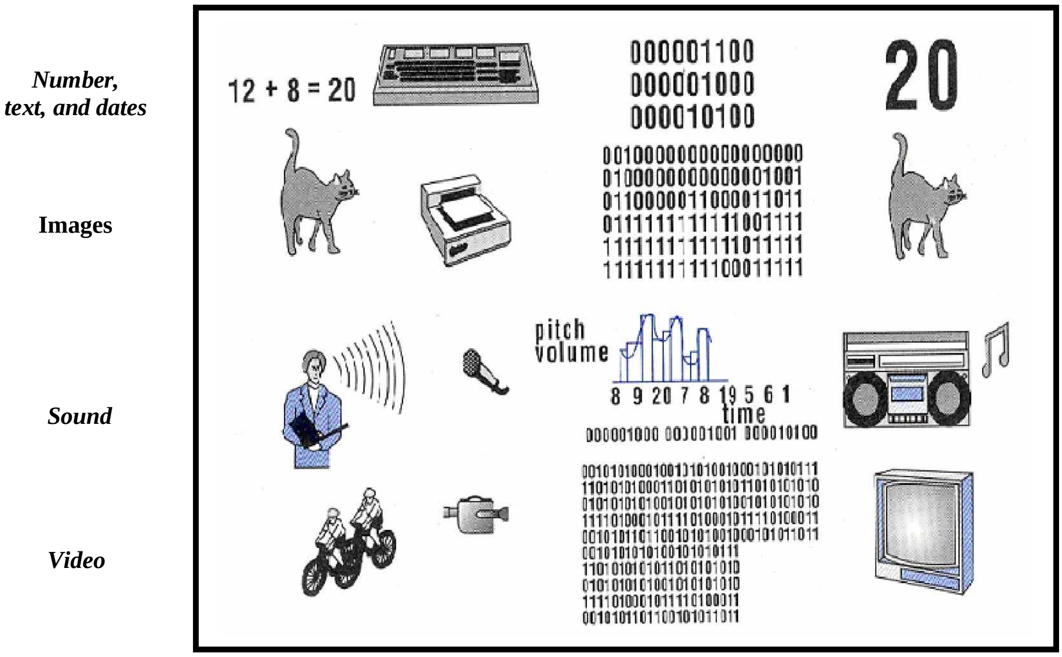
2. Data Types and Objects

The fundamental data types are shown in Figure (1). Usually before the computer can store and process data, it must be reduced by input devices to binary format. Output devices return the binary data to a form that a humans can handle. Each data type can have many different subtypes. For example, numbers can be classified as integer or real. Text may look simple, but it become more complex when you need to store alphabets from different languages. Date and time data can be considered as text, but it is best to treat it as a separate data type. Images can be stored in any of a hundred different formats, split between bitmap and vector images. Similarly, there are many ways to store sound and video clip.

Handling complex objects presents two basic difficulties:

i- They require considerably more space. For example , a detailed image can easily take several megabytes of storage space. Sound and video need huge amounts of space because every second generates more data. As a example, even with compression, a high-resolution video clip with sound can fill 2 gigabytes of storage in 10 minutes.

ii- The second difficulty presented by complex objects is that they cannot be manipulated by traditional commands. For example, there is no obvious way to sort a table of sound clips. Yes you can sort on other attributes, such as length or volume, but comparisons of the sound itself is difficult[1]



Figure(1) [1] basic data objects, and Complex data objects

3. Relational database

A relational database stores all its data inside tables and nothing more. All operations on data are done on the tables themselves or produce another tables as the result. You never see anything except that tables. A table is a set of rows and columns and a set does not have any predefined sort order for its elements. Each row is a set of columns with only one value for each. All rows from the same table have the same set of columns, although some columns may have NULL values, i.e. the value for that row is not initialized. It is to be noted that a NULL value for a string column is different from an empty string.

As an example, the Relational model[2] supports relations, which are set of tuples with fixed number of primitive data elements. The rows from a relational table are analogous to a record and the columns to a field. Here's an example of a table and the SQL statement that creates the table:

```
CREATE TABLE ADDR_BOOK
NAME char(30),
COLLEGE  char(20),
E_MAIL char (25) )
```

Name	**College**	**e-mail**
Yahya mahdi hadi Al-mayali	**Computer Science & Math.**	**yahyamh@edu.kuiraq**
Hussian Mohi Al-Musawi	**Kufa University**	**hussianma@edu.kuiraq**

There are two basic operations that we can perform on above relational table. Retrieving a subset of its columns and retrieving a subset of its rows. Here are samples of the two operations:

```
SELECT NAME, E_MAIL FROM ADDR_BOOK
```

The result is:

```
      NAME                         E_MAIL
Yahya mahdi hadi Al-mayali      yahyamh@edu.kuiraq
Hussian Mohi Al-Musawi         hussianma@edu.kuiraq
```

```
SELECT * FROM ADDR_BOOK WHERE COLLEGE =" Kufa University"
```

The result is:

Name	**College**	**e-mail**
Hussian Mohi Al-Musawi	**Kufa University**	**hussianma@edu.kuiraq**

We can also perform operations between two tables treating them as sets: we can make Cartesian product of the tables and can get the intersection between two tables, we can add one table to another and so on. Later we should be discussing these operations in OODBMS and show how they are more useful and better.

4. Object oriented databases

In this paper, we examine object systems by introducing and explaining basic object oriented concepts and offer some opinion regarding the suitability of incorporating such concepts into the database systems of the future. The advent and commercial success of well-engineered ODBMS products, such as ObjectStore[3], indicate that the time is ripe to seriously investigate migration from RDBMS to ODBMS.

The classical SQL systems being inadequate in a variety of ways, we are led to study object systems.

5. 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-oriented 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.

All of these applications are characterized by having to manage complex, highly interrelated information, which is the strength of object-oriented database systems. Clearly, relational database technology has failed to handle the needs of complex information systems. The problem with relational database systems is that they require the application developer to force an information model into tables where relationships between entities are defined by values.[7]

6. Comparison between relational and object-oriented databases

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 are possible. Object database design is quite different. For the most part, object database design is a fundamental part of the overall application design process. [6]

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[5].

An initial area of focus by several object-oriented database vendors has been the Computer Aided Design (CAD), Computer Aided Manufacturing (CAM) and Computer Aided Software Engineering (CASE) applications. A primary characteristic of these applications is the need to manage very complex information efficiently. Other areas where object-oriented database technology can be applied include factory and office automation. For example, the manufacture of an aircraft requires the tracking of millions of interdependent parts that may be assembled in different configurations. Object-oriented database systems hold the promise of putting solutions to these complex problems within reach of users.

Object-orientation is yet another step for expressing solutions to problems in a more natural, easier to understand way. fundamental characteristic of the new level of system description is that it is closer to the human conceptualization of a problem domain". Descriptions at this level can enhance communication between system designers[4].

Object-oriented concept:

The object-oriented paradigm is the latest in the software development and the most adopted one in the developing project of today. RDBMS extensions have been spurred by competition from object-oriented database management systems (ODBMSs), which combine comprehensive database management functionality and full-fledged OO data modeling[3].

Limitation of Procedural Programming: A Program in a procedural language is a list of instructions where each statement tells the computer to do something. The focus is on the processing, the algorithm needed to perform the desired computation.

* In procedural paradigm, the emphasis is on doing things. And not on the data. But Data is, after all, the reason for a program's existence. The important part of an inventory program isn't a function that display or check data; it is the inventory data itself. Yet data is given second – class status while programming.

* In procedural programming, data type are used and worked upon by many functions. If a function makes any change to a data type, then it must be reflected to all the locations, within the program that process this data type. This is very time consuming for large sized programs.
* Procedural programming does not model real world very well. For instance, a vehicle is an object, which is capable of moving in real world. However, the procedural programming paradigm would just be concerned about the procedure i.e. the procedure programming paradigm would just think of moving the part and not the vehicle.

7. Object Oriented Analysis

Object Oriented Analysis (OOA) views the world as objects with data structures and behaviors and events that trigger operations (object behavior changes). This in turn changes the state of objects. The idea that a system can be viewed as a collection of interacting objects, each of which is a batch of data and functionality, is the foundation of this technology and provides an attractive alternative for the development of complex systems. This is a far-reaching departure from prior methods of requirements specification, such as structured analysis and design and functional decomposition.

We all know what objects are. An object is a representation of a real-life entity or abstraction. For instance, objects in a car rental system might include: a car, route, an icon on a screen, or even a full screen with which a travel agent interacts. The OOA is used to specify the structure and behavior of the object i.e. the requirements of that object. To specify the requirements of the objects different types of models are required. The information or object model will contain the definition of objects in the system, including: object name, object attributes, and object relationships to other objects. This technology works best when used in new development. OO Analysis involves the team to arrive at the description of the problem and subsequently identify the requirements. The fundamental question to ponder over here is what the problem is about and what a system must do. Analysis emphasizes and investigates the problem rather than how a solution is desired. Once this is done, it’s time to move on to the next phase i.e. the design part.

8. Object Oriented Design

Object-oriented design (OOD) is mainly concerned with developing an object-oriented model of a system to implement the identified requirements. It involves arriving at detailed descriptions of logical solution to the problem identified during the OOA stage. It seeks answers to questions like how the logical solution fulfills requirements and constraints. It builds on the products developed during OOA by refining candidate objects into classes, defining data structures and procedures, and defining message protocols for all objects. the most important part of OOD is mapping the above mentioned things into an object-oriented programming language (OOPL). OOD requires the specification of concepts that are not existent in analysis, such as the types of the attributes of a class, or the logic of its methods. Design can be thought of in two phases: The first phase is called the high-level design, deals with the breaking up of the system into large, complex objects. The second phase is called low-level design. In this phase, attributes and methods are specified at the level of individual objects. This is also where a project can realize most of the reuse of object-oriented products, since it is possible to guide the design so that lower-level objects correspond exactly to those in existing object libraries or to develop objects with reuse potential.

OOD techniques are useful for development of large complex systems. It can yield the following benefits: maintainability through simplified mapping to the problem domain, which provides for less analysis effort, less complexity in system design, and easier verification by the user; reusability of the design artifacts, which saves time and costs; and productivity gains through direct mapping to features of Object-Oriented Programming.

The use of object oriented technology requires formal training in OOA methods. A method of training that has produced desired results at Stylus is to initiate pilot projects, conduct formal classes, and conduct team reviews to properly train all the analysis and development staff as well as the program management team. Technical management at Stylus Systems is aware that the maximum impact from OOAD is achieved when used with the goal of designing reusable software. Another very important knowledge is that for objects without significant reuse potential, OOAD techniques were more costly than traditional software development methodologies.

Like most database applications, object databases are subject to evolution. Evolution, however, is critical in object oriented databases since it is the very characteristic of complex applications for which they provide inherent support.
These applications not only require dynamic modifications to the data residing within the database but also dynamic modifications to the way the data has been modeled (i.e. both the objects residing within the database and the *schema* of the database are
subject to change). Furthermore, there is a requirement to keep track of the change in case it needs to be reverted. Object database schemata designed to fulfill the above set of requirements can become very large and complex. The large amount of information and complex relationships between the various entities in these schemata combine to make the process of assessing the effect of change expensive, time consuming and error-prone. However, without proper assessment, it is impossible for developers and maintainers to fully appreciate the extent and complexity of proposed changes[7].

9. An overview of object technology

It is a basic concept of the Object approach that “everything is an object”. Some objects are fixed; examples might be integer (15,60) and character string (“Najaf”, “Baghdad”). Other objects are variable; examples might be the department
and employee.

Objects are encapsulated, which means that the physical representation i.e. the internal structure of such an object, say a Dept (“department”) , is not visible to
users of that object; instead, user knows only that the object is capable of executing certain operations (Methods).

10. Creation of object oriented database

Assume we wish to define two object classes namely DEPT (departments) and EMP (employees). Also suppose that the user-defines classes MONEY and JOB and the class CHAR is built-in. Then the necessary class definition for DEPT and EMP might look somewhat as follows:

```
CLASS DEPT
PUBLIC (Dep# Char, Dname Char, Budget Money, MGR REF(EMP),
EMPS REF(SET(REF(REF(EMP))))----
```

```
METHODS (HIR_EMP(REF(EMP))---code----,
FIRST_EMP(REF(EMP))—code----,----‘
CLASS EMP
PUBLIC (EMP# CHAR
ENAME CHAR
SALARY MONEY
POSITION REF (JOB))---
METHOD (----)---;
```

Object-orientation is yet another step in the quest for expressing solutions to problems in a more natural, easier to understand way. Michael Brodie in his book On Conceptual Modeling[6] states, "The fundamental characteristic of the new level of system description is that it is closer to the human conceptualization of a problem domain”. Descriptions at this level can enhance communication between system designers.

Object-oriented concept:

The object-oriented paradigm is the latest in the software development and the most adopted one in the developing project of today. RDBMS extensions have been spurred by competition from object-oriented database management systems (ODBMSs), which combine comprehensive database management functionality and full-fledged OO data modeling[7].

Limitation of Procedural Programming: A Program in a procedural language is a list of instructions where each statement tells the computer to do something. The focus is on the processing, the algorithm needed to perform the desired computation.

* In procedural paradigm, the emphasis is on doing things. And not on the data. But Data is, after all, the reason for a program’s existence. The important part of an inventory program isn’t a function that display or check data; it is the
inventory data itself. Yet data is given second – class status while programming.

* In procedural programming, data type are used and worked upon by many functions. If a function makes any change to a data type, then it must be reflected to all the locations, within the program that process this data type. This is very time consuming for large sized programs.

* Procedural programming does not model real world very well.

For instance, a vehicle is an object, which is capable of moving in real world. However, the procedural programming paradigm would just be concerned about the procedure i.e. the procedure programming paradigm would just think of moving the part and not the vehicle.

OO programming: Now, the object oriented approach views a problem in terms of objects involved rather than procedure for doing it.

Object: object is an identifiable entity with some characteristics and behavior. For instance, we can say ‘Orange’ is an object. Its characteristics are: It is spherical shaped, its color is Orange etc. Its behavior is: it is juicy and it tastes sweet sour. While using OOP approach the characteristics of an object are represented by its associated functions. Therefore, in Object Oriented Programming object represents an entity that can store data and has its interface through function[5].

Everyday uses of object databases: We can use object
database in the following:

* Pager
* Voicemail
* Flight booking

* PCs phone

Object databases are used more often than we might realize. Many times, using an object database is seen as competitive advantage and companies do not want to publicize this. As a result, object databases are invisible to users and not mentioned by companies and hence do not receive much media attention

High performance: With complex data, it is not unusual to find that an 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.

Why are ODBMSs faster? ODBMSs are optimized for the traversals related to complex data. They also do not have any “ impedance mismatch” when it comes to

11. Conclusion

1. In this study we have focused on migration from RDBMS to ODBMS. We have also discussed that ODBMS is better and faster than RDBMS for complete data.
2. The Object-Oriented Approach can store and process the complex data types such as: Graphics, Sound, and Video clips in addition to basic data types.
3. The long-term solution thus may be a hybrid system, in which the ODBMS manages the live data, which is flushed to the RDB when data processing is required.

12. Refrences

[1] Gerald V. Post, 2002. "Database Management Systems", McGraw-Hill Higher Education.

[2] Date, C.J., 1985. "An Introduction to Database System", Addison Wesley.

[3] Brodie, M., J. Mylopoulos and J. Schmidt, 1985. On Conceptual Modeling. Springer-Verlag.

[4] Michael, S. and G. Kemnitz, 1991. The postgres next generation database management system. Commun. ACM, 34: 78-92.

[5] Arthur, M.K., R. Jensen and S. Agarwal, 1993. Persistence software: Bridging object-oriented programming and relational databases. Proc. ACM SIGMOD Intl. Conf. Management of Data, Washington DC, pp: 523-528

[6] Bahrami, A. Object-Oriented System Development. New York: McGraw-Hill,1999.

[7] http://www.postgreSQL.org , (Object-Relational DBMS).

أسلوب التقنية الشيئية لقواعد البيانات تمثل الحل الأمثل لنظم التطبيقات المتكاملة

يحيى مهدي هادي الميالي

قسم علوم الحاسبات/كلية الرياضيات وعلوم الحاسبات/ جامعة الكوفة

الخلاصة

أكثر نظم قواعد بيانات العمل الحالية تنظم البيانات على شكل أعمدة من الأرقام أو النصوص القصيرة أو التواريخ . لكن التطــور السريع في إمكانيات وقدرات الحاسبات الرقمية جعل كثير من مستخدمي الحاسبات يتطلع إلى امتلاك نظم قواعد معلومات لها إمكانية التعامل مع عناصر كيانيه معقدة مثل: إشكال مصورة (Graphics)، أصوات (Sounds)، أو أفلام مصورة (Video Clips)، بعض التطبيقات مثل

نظام المعلومات الجغرافية، وبرمجيات استخدام الحاسوب للمساعدة على التصميم يحتاج المصمم إن يستخدم بعض الكيانات الخاصة بكل تطبيق لذلك أصبح مصممي النظم والتطبيقات بحاجة ملحة لوجود تقنيات برمجية فيها الإمكانية لتعريف بعض الكيانات الخاصة ولأ غراض مختلفة ضمن تطبيق واحد إن عملية خزن الكيانات يتطلب عمليتين أساسيتين : خزن البيانات لخواص الكيان وخزن وتفعيل طرق معالجة بيانات الكيان.

لقد تم تطوير برمجيات قواعد البيانات العلائقية لكي تتعامل مع الكيانات(بيانات وخواص). إن احد العناصر الأساسية التي يمتلكها هذا الأسلوب هي القدرة على تعريف عناصر جديدة للبيانات، تعامل هذه العناصر الجديدة ككيانات ويمكن لها إن ترث خواص من أنواع بيانات أخرى. إن وجود عناصر بيانات جديدة يمكن لمصممي التطبيقات من تنفيذ تطبيقات متقدمة ومتكاملة ويمكن إن تتضمن جميع خواص المستندات الموزعة (Spreadsheet)، معالج النصوص (Word processor)، وكذلك برمجيات الرسوم (Graphic Package). يمكن لمصمم النظم إن يبني أي نظام معقد من خلال ربط الكيانات المختلفة مع بعضها البعض.

في هذا البحث سنتطرق إلى الجوانب الايجابية التي يمتلكها موضوع بناء قواعد الـ بيانات الكيانية مع المقارنة بنظم قواعد البيانات العلائقية.