UDC 681.3.07

P. ANZENBERGER[1,2], A. CHUKHRAY[1]

[1]National Aerospace University 'Kharkiv Aviation Institute' , Kharkiv, Ukraine
[2]University of applied Sciences, Wels, Austria

OBJECT-RELATIONAL DATA MODELS FOR REAL THE WORLD

Object oriented programming became very popular in the last years. The idea to imitate real-world patterns (building systems with interacting objects) in system modeling and software development is yet as simple as powerful. Without raising fundamental discussions on pros and cons of OOP in general, it can be stated that this concept has grown to large popularity and usability nowadays.

relational model, object model, object-relational model

Introduction

Object oriented programming became very popular in the last years. The idea to imitate real-world patterns (building systems with interacting objects) in system modeling and software development is yet as simple as powerful. Without raising fundamental discussions on pros and cons of OOP in general, it can be stated that this concept has grown to large popularity and usability nowadays.

When looking at the field of database technology, the degree of object orientation is surprisingly small. This founds in various reasons, one of which is that the „classical“ approach to represent data as sets of related tuples (which is called „relational data“) has a long and successful history in information technology. The used associations are simple and often yield a rather „flat“ hierarchy. Relational database management systems (RDMBSs) are today's most popular and widely used database systems.

In OOP, it is typically the behavior of objects (use cases, algorithmic logic) being emphasized. On the other hand, it is the data that counts in database technology. This fact serves as a common motive for the combination of these two paradigms.

This combination can be seen in the object-relational databases.

1. What are object-relational databases?

An object-relational database (ORD) or object-relational database management system (ORDBMS) is a relational database management system that allows developers to integrate the database with their own custom data types and methods. The term *object-relational database* is sometimes used to describe external software products running over traditional DBMSs to provide similar features; these systems are more correctly referred to as object-relational mapping systems.

Whereas RDBMS or SQL-DBMS products focused on the efficient management of data drawn from a limited set of data types (defined by the relevant language standards), an object-relational DBMS allows software developers to integrate their own types and the methods that apply to them into the DBMS. The goal of ORDBMS technology is to allow developers to raise the level of abstraction at which they view the problem domain.

2. Why object-relational databases?

The developers of database applications think today in terms like “Entities”, “Tables” and “Columns”. This may be good, but the disadvantage is, they can only use the features the database management system is offering. Every other aspect has to be implemented outside the database, although they like to implement a lot inside the database. We know many applications with many stored procedures, triggers and views. And there is always a conflict between the application developer and the database developer which functionality where to implement.

Mismatch Category	Object Approach Representation	Relational Database Representation
Main Scope	Behavioral	Data-Centered
Identification	Object Identity	Primary Keys
Associations	Pointers / References	Foreign Keys
Data / Object Structure	Network of Objects	Rows
Data / Object Retrieval	Traversing Relationships	Queries
Information Hiding	Encapsulation	Program Independence

In [1], the term “impedance mismatch” is explained as follows:

“Of the primary object-modeling properties, relational modeling seems to have no way of representing any of them directly. Tuples have neither identity nor encapsulation. Tuple attribute values are encapsulated but are pure values, so they have neither identity nor state. This is what is frequently called an impedance-mismatch between object approaches and relational databases.”

The features of object relational database systems could solve some problems the relational database has. That’s why the data not only have values, but also have a behavior now.

3. Basic concepts for ORDBMS

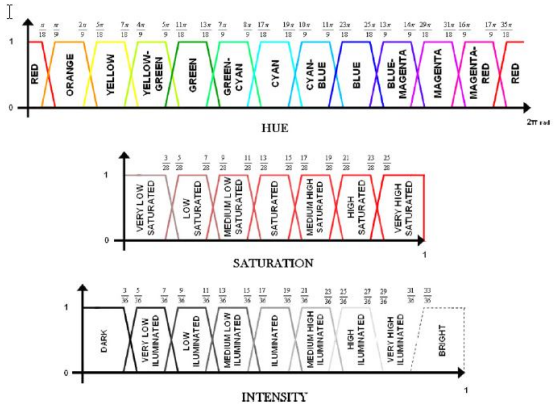
- Object Identity and References
  - Objects own an unequivocal Object identifier (OID)
    - OID is independent of the objects value
  - Object references are OIDs
  - Identical objects have the same OID; the same objects have the same object value
- Complex Structures (data types)
  - Created by the use of nested type constructors
    - Tuple
    - Sets, Multi Sets, Lists, Arrays
    - References
  - nested, navigating inquiries by means of path expressions
- Encapsulation
  - Separation of interface and implementing (ADT draft)
  - Object access only about methods of the interface
- Types versus Classes (object tables)
  - type describes structure of an amount from objects
  - class offers extension (container) for objects of a type
- Type Hierarchy and Class Hierarchy (Object Table Hierarchy)
  - sub type inherits type (attributes and methods) of a super type
  - super class also contains all authorities of the sub classes
- Overload and Late Binding
  - Redefinition (Overriding) of methods in sub types
  - Name of the method stays the same, nevertheless, the effect is dependent on object
- Extensibility
  - user-defined types can be introduced in the system
  - Predefined base types and user-defined types can be combined

Examples for this are:

Fuzzy search

Because the object has not only data but also a behavior, the object can answer some questions by itself.

Fuzzy search of colors [2]



So you can query the color of an object in a fuzzy way and not like now with exact values.

```
CREATE TYPE color AS
(hue              NUMBER,
 saturation       NUMBER,
 intensity        NUMBER)
 METHOD IsRed() RETURNS BOOLEAN,
 METHOD IsLowSaturated() RETURNS BOOLEAN,
 METHOD IsDark() RETURNS BOOLEAN,
. . .
 METHOD SetToRed() RETURNS color,
 METHOD SetToDark() RETURNS color,
 . . .
);
```

Strong typing

Before SQL99, columns could only be defined with the existing built-in data types. There was no strong typing so logically incompatible variables could be assigned to each other.

```
CREATE TABLE RoomTable (
RoomID          VARCHAR(10),
RoomLength              INTEGER,
RoomWidth               INTEGER,
RoomArea                INTEGER,
RoomPerimeter   INTEGER);
UPDATE RoomTable SET RoomArea = RoomLength;
→ No Error Results
```

In an object relational data model each UDT is logically incompatible with all other types:

```
CREATE TYPE roomtype AS CHAR(10) FINAL;
CREATE  TYPE meters AS INTEGER FINAL;
CREATE  TYPE squaremeters AS INTEGER FINAL;
CREATE TABLE RoomTable (
RoomID                  roomtype,
RoomLength              meters,
RoomWidth               meters,
RoomPerimeter   meters,
RoomArea                squaremeters);
UPDATE RoomTable SET RoomArea = RoomLength;
→ ERROR
UPDATE RoomTable SET RoomLength = RoomWidth;
→ No Error Results
```

4. Measurements for Environmental Information Systems

Especially in measurement systems the behavior of data is extremely important. For example an environmental information system must fulfill not only technical requirements but also a lot of restrictions in respect of local and international conventions and laws.

Not only the security of the data is very important, but also the methods of manipulating this data is regulated by laws. Therefore it is very important, that the logic is no longer in applications but isolated in the database. This can be realized by the use of object relational databases.

```
CREATE TYPE MeasurementType as
(measuring_date DATE,
 measuring_value          NUMBER);
CREATE TYPE MeasurementTableType AS TABLE
OF MeasurmentType;
CREATE TYPE HalfHourAvgValue AS
(LocationID              INTEGER,
 AvgValue                NUMBER,
 CountValues             INTEGER,
 Measurements    Measurement Table Type)
FINAL
 METHOD AddMeasurement (Measurement
MeasurementType),
 METHOD GetAvgValue() RETURNS NUMBER,
 . . .
);
```

By implementing the logic within the database the developer can control how the data can be manipulated. The database can protect the data from illegal manipulation, not only by means of the security and access restriction of the database management system but also by encapsulating the data. It is not possible to change the data of the measurements or of the AvgValue because only the object itself can do these operations. The access of the real data itself is not possible.

References

1. Fussell, Mark L.: Foundations of Object Relational Mapping, v0.2 [mlf- 970703], published online at http://www.chimu.com, copyright by Mark Fussell, 1997

2. Retrieving images in fuzzy object relational databases using dominant color descriptors From J.Chamorro-Martínez, J.M.Medina, C.D.Barranco, E.Galán-Perales, J.M.Soto-Hidalgo Department of Computer Science and Artificial Intelligence, University of Granada, C/Periodista Daniel Saucedo Arandas, 18071 Granada, Spain Polytechnic, Pablo de Olavide University, UtreraRd.Km.1, 41013 Seville, Spain.

Поступила в редакцию 28.01.2007

Рецензент: лауреат Государственной премии Украины, д-р техн. наук, проф. А.С. Кулик, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.