Migration of Responses of Questionnaires from MS-Excel into OODBMS

Dr. Ranjana Ingolikar[1]	**Prof. Rasika Khandal[2]**
Dept. of Computer. Science.	*MCA Department*
SFS College - Nagpur	*SRPCE - Nagpur*
RTM Nagpur University, India	*RTM Nagpur University, India*

Abstract — The purpose of this paper is to identify the way of migrating of responses of questionnaires from MS-Excel into the object oriented data model. To explore the way of migrating responses which is stored in MS-Excel datasheet, help has been taken of a questionnaire designed by a business management researcher. The simple excel response sheet is migrated into OODBMS by using object oriented programming language terminology and save the data of responses into the object oriented database. A comparative study of storage of responses of questionnaires in MS-Excel and OODBMS databases is made. It has been found that the database OODBMS is a better option for storing responses of questionnaires.

Keywords — Questionnaire, Object Oriented database System (OODBMS), Immutable objects, Mutable objects.

I. INTRODUCTION

A. Questionnaire

A questionnaire is one type of data collection method, which contains formalized set of structured type of questions and to obtain responses from respondents. Respondents were asked to mark their responses. Usually novices store their responses in the structured excel sheet. Here an attempt is made to associate relational data model with the storage of questionnaire responses. Business Management researcher whose case study was considered for research had used excel response sheet for storing responses. It is redefined by using relational data model approach.

B. Object Oriented Database System

An object oriented database management system is the result of combining object oriented programming principles with database management principles. Object oriented programming concepts such as encapsulation, polymorphism and inheritance as well as database management concept such as ACID properties (Atomicity, Consistency, Isolation and Durability) are enforced which lead to system integrity. It supports an ad hoc query language and secondary storage management system which allows managing of vary large amount of data. It supports the following features as mandatory for a system it can be called as OODBMS; Complex Object, 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. In the object oriented database model, any real world entity is represented by only modelling concept – the object. An object has a state and behaviour associated with it. The state of an object is defined by the value of its properties (attributes). Properties can be primitive values (like strings and integers) and non-primitive objects. The behaviour of an object is specified by methods that operate on the state of the object. Each object is uniquely identified by a system-defined identifier (OID). Objects with the same properties and behaviour are grouped into classes. An object can be an instance of only one class or an instance of several classes. Data definition and data manipulation language (DDML) allows persistent data to be created, updated, deleted or retrieved. DDML allows defining a database, maintaining and querying a database including updating, inserting, modifying, and querying data by using class methods [1]. Here it is proposed that the object oriented model design can be used to partition the questionnaire into different classes where each class supports the object oriented programming concept. To illustrate the concept of partitioning, the following case study has been considered.

II. CASE STUDY

The management researcher Mrs. Soma Sharma, Research scholar of RTM Nagpur University, in her Ph.D. thesis "A Study of Micro Finance as a tool for alleviating urban poverty with reference to Nagpur City" has done her work in MS-Excel response sheet for storing responses.

In this paper the best way of storing given responses in the object oriented data model is proposed. The questionnaire contains 35 structured questions which are multiple choice questions filled from respondents and whose responses were stored in excel which are to be partitioned into classes using OOBMS. It is used to store data in form of objects.

- *Multiple-Choice Questions:* In which the management researcher provides a choice of answers and respondents are asked to select only one alternative from the given choices.

A. Problems/Limitations of MS-Excel sheet

The researcher used MS-Excel sheet for storing responses. With reference to the case study considered for research, storing responses in the MS-Excel datasheet is not an efficient way. It has the following major limitations:

1) Data types and default values of data types are not supported in excel.
2) Field width or size of field is not supported in excel.
3) Validation rules and validation text cannot be applied.
4) Text formatting is not supported.
5) MS-Excel does not give the facility of data query processing.
6) Data present in an MS-Excel does not support referential integrity rules for establishing relationship between data.
7) There are limitations in data storage size. Eg. In the new version of MS-Excel 2007, the limitations are 1,048,576 rows by 16,384 columns. [a][b].

Looking at the above limitations or problems of MS-Excel datasheet. The proposed partitioning of questionnaire and migrate the MS-Excel datasheet into the object oriented data model for storing responses.

B. Partition of Questionnaire

The given MS-Excel datasheet contains the responses of questionnaires and it looks as Fig. 1:

Fig. 1 MS-Excel responses datasheet of MicroFinance

Object oriented approach views a problem in terms of objects rather than procedure. Object is an identifiable entity with some characteristics and behaviour. Object is an instance of a class. For identifying object class and their associate member variables or methods the following general model is used by user for storing responses of questionnaires.
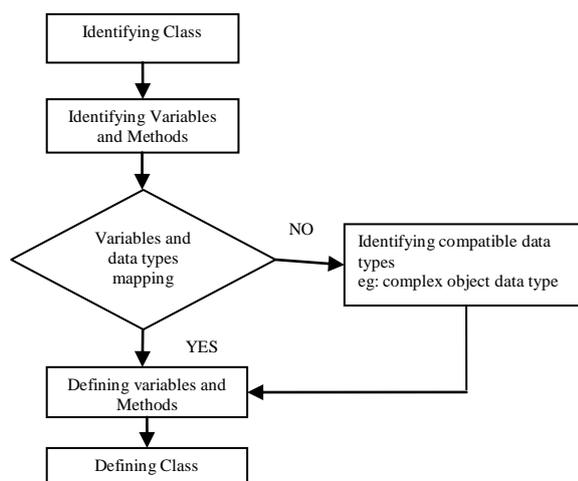

Fig. 2 General Model of OODBMS

By using the above fig. 2 it can be concluded that in order to store the responses of respondent to the questionnaire the following entities or classes are used:

1) RESPONDENT_DETAILS
2) RESPONDENT_FAMILY_DETAILS
3) MICRO_FINANCE

4) SHG
5) SHG_ LOAN

To illustrate this concept by creating an application in object oriented programming language C#.net and to store the responses of questionnaires objects in object oriented database. The definition of class hierarchy in C#.net is as below:

1) *CLASS RESPONDENT_DETAILS*

```csharp
public class RESPONDENT_DETAILS
{
    int RSPD_No;
    string RSPD_NM;
    int RSPD_Age;

    //constructor
    public RESPONDENT_DETAILS(int RSPD_Number, string RSPD_Name, int Age)
    {
        RSPD_No = RSPD_Number;
        RSPD_NM = RSPD_Name;
        RSPD_Age = Age;
    }

    //get property
    public int RSPD_Number
    {
        get
        {
            return RSPD_No;
        }
    }

    public string RSPD_Name
    {
        get
        {
            return RSPD_NM;
        }
    }
    public int Age
    {
        get
        {
            return RSPD_Age;
        }
    }
}
```

2) *CLASS RESPONDENT_FAMILY_DETAILS*

```csharp
public class RESPONDENT_FAMILY_DETAILS : RESPONDENT_DETAILS
{
    int No_Fm_Mem;
    int No_Earning_Mem;
    long Monthly_Income;

    //constructor
    public RESPONDENT_FAMILY_DETAILS(int FmilyMembers, int EarningMembers, long
MonthlyIncome,int   RSPD_Number, string RSPD_Name, int Age): base(RSPD_Number ,RSPD_Name ,Age )
    {
        No_Fm_Mem = FmilyMembers ;
        No_Earning_Mem = EarningMembers ;
        Monthly_Income = MonthlyIncome ;
    }

    //get property
    public int FmilyMembers
```

```
                {
                        get
                        {
                            return No_Fm_Mem;
                        }
                }

                public int EarningMembers
                {
                        get
                        {
                            return No_Earning_Mem;
                        }
                }
                public long MonthlyIncome
                {
                        get
                        {
                            return Monthly_Income;
                        }
                }
            }
        }
```

3) CLASS MICRO_FINANCE

```
    public class MICRO_FINANCE
    {
        string MF_aware;
        string MF_References;
        RESPONDENT_DETAILS respdtdetail;

        //constructor
        public MICRO_FINANCE(string MFaware, string MFref, RESPONDENT_DETAILS respondenttdetail)
        {
            MF_aware = MFaware;
            MF_References = MFref;
            respdtdetail = respondenttdetail;
        }

        public RESPONDENT_DETAILS respondenttdetail
        {
            get
            {
                return respdtdetail;
            }
        }
        public string MFaware
        {
            get
            {
                return MF_aware;
            }
        }
        public string MFref
        {
            get
            {
                return MF_References;
            }
        }
    }
```

4) CLASS SHG

```
public class SHG : MICRO_FINANCE
{
    string SHG_Mem_Status;

    public SHG(string SHGMemstatus, string MFaware, string MFref, RESPONDENT_DETAILS
    respondtdetail)  : base(MFaware, MFref, respondtdetail)
    {
        SHG_Mem_Status = SHGMemstatus;
    }

    public string SHGMemstatus
    {
        get
        {
            return SHG_Mem_Status;
        }
    }
}
```

Look at the above class definition, class hierarchy is formed between the classes of entities. OODBMS supports inheritance that means using parent-child relationship concepts to access all public properties by the child class. Also it supports complex data structure **For example:** to create object data type variable **'respdtdetail'** of type RESPONDENT_DETAILS. By using the object oriented programming language, creates class hierarchy for all entities and stores the responses of questionnaires in object oriented database [2].

III. AN OVERVIEW OF OBJECT TECHNOLOGY

It is a basic principle of the Object approach that "everything is an object". Some objects are immutable*; examples might be integer and string. Other objects are mutable*; example might be MicroFinance.

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

A. Persistence of objects

Persistence means that certain program components "survive" the termination of the program. Thus these components have to be stored permanently on the secondary storage device. Typically, persistence or non-persistence is specified at object creation time. There are two possible ways to make an object persistent:

1) Explicitly call built-in function persistence. Certain objects are persistent.
2) Automatically make object of persistent types persistent. All objects are persistent.

There are several object-oriented DBMS products in the market (e.g., db4o, Gemstone, Objectivity/DB, ObjectStore, Ontos, O2, Itasca, and Matisse). These products all support an object-oriented data model. Specifically, they allow the user to create a new class with attributes and methods, have the class inherit attributes and methods from super classes, create instances of the class each with a unique object identifier, retrieve the instances either individually or collectively and load and run methods using DDL and DML. These products have ability to directly manipulate data stored in a database using an object oriented programming language. Using an object oriented database product means that you have higher performance and less code to write [3].

B. How this approach served the purpose of applying this method

In this method we follow the two steps:

1) Partitioning the questionnaires into the number of classes.
2) Storing the responses of questionnaires into OODBMS.

The above approach is used to decompose the given MS-Excel datasheet into number of classes which serves the object oriented technology. Ones the classes are created it is possible to store the partitioned data in OODBMS in the form of objects. For storing responses of questionnaire, we used object oriented database system which gives advantage in storing complex data type and query processing.

IV. BENEFITS OF OODBMS DATA MODEL

When questionnaire data stored in OODBMS, it gives better performance than MS-Excel datasheet. It handles complex data type like object data type, to support all object oriented programming concepts like inheritance, polymorphism etc., and high level of query processing engine. Some benefits of object oriented database management system are given below:

1) Data can be stored in the form of object.
2) Because of the OOP's, only exact and accurate information can be fetched.
3) In OODBMS, data is described easily through class hierarchy.
4) OODBMS support functionally of OOPs using the combination of object oriented programming language with database technology.

5) It supports all data types with their default values and field width.
6) OODBMS supports complex object type such as graphics, audio/video or geographic information.
7) Validation rules and validation text can be applied on given member variable of class.
8) Inheritance is used to established parent-child relationship between classes.
9) There is no physical limit for storing of number of objects in database files.
10) Unique object identification (OID) is used to communicate with other objects.
11) OODBMS is better suited to handle complex, interrelated data than MS-Excel.
12) OODBMS support object oriented programming concepts such as encapsulation, inheritance, polymorphism, dynamic binding etc.
13) Query processing is not necessary, still DDL and DML used to querying data from database by using powerful query engine.

V. PROBLEMS/LIMITATION OF OODBMS DATA MODEL

1) OODBMS is language dependent because it tied to a specific language.
2) Complexity: OODBMS is more complex than MS-Excel. It is more expensive and difficult to use.
3) Data cannot easily export from OODBMS to MS-Excel.

VI. CONCLUSIONS

From the above case study we conclude that:
1) OODBMS data model is a better fit in storing response of questionnaire data as compare to MS-Excel datasheet.
2) Use of OOP's concept guarantees that the originality of the response sheet is maintained.
3) Storing data in object data model and defining class hierarchy between entities will produce the transactional data which proves to be easy to handle.
4) Message passing between objects can be easily handled Using parent-child relationship. It is not supported by MS-Excel.
5) Query processing is not necessary in OODBMS, but still it possible to execute queries fast using high query processor engine which is not supported by MS-Excel.
6) To define data types for each member variable of class with their size this is not possible in case of MS-Excel.
7) In OODBMS for each member variable of class with their size can be defined but this is not possible in case of MS-Excel datasheet.
8) OODBMS support object oriented programming concepts such as encapsulation, inheritance, polymorphism, dynamic binding etc. but it is not possible in MS-Excel datasheet.
9) Complex data can be easily stored in OODBMS which is not supported by MS-Excel.

ACKNOWLEDGMENT

REFERENCES

[1] T. Atwood, *"An object-oriented DBMS for design support applications"*, Ontologic Inc. Report.
[2] D. Maier, J. Stein, A. Otis, A. Purdy, *"Development of an object-oriented DBMS"* Report CS/E- 86-005, Oregon Graduate Center, April 86.
[3] Malcolm Atkinson, François Bancilhon, David DeWitt, Klaus Dittrich, David Maier, Stanley Zdonik (1989), *"The Object-Oriented Database System Manifesto"*.

REFERENCE SITES

[1] http://spreadsheetpage.com
[2] http://wiki.answers.com/Q/Total_number_of_row_and_column_of_m.s.excel