

Design and Implementation of Probabilistic Assignment Management System for a Library Database Using Object Oriented Programming

Jamal M. Hameed

Scholarship & Cultural Relations Directorate
Baghdad - IRAQ

Received on: 15/06/2011

Accepted on: 03/10/2011

المخلص

في السنوات الأخيرة أصبحت نظم المعلومات المحوسبة واسعة الانتشار مما جعلها أداة مهمة في مختلف حقول الحياة. من أهم التطبيقات لنظم المعلومات هي استخدامها في إدارة المؤسسات التجارية وحتى الأعمال الصغيرة, لذلك نستطيع القول انه ليس هناك إدارة ناجحة في مختلف الأعمال بدون أنظمة المعلومات. يهدف هذا البحث إلى تطبيق البرمجة الكيانية لتصميم وتنفيذ نظام إدارة التخصيص الاحتمالي لقاعدة بيانات مكتبة تجارية. إن معظم تطبيقات البرمجة الكيانية استخدمت لغات البرمجة الكيانية, وفي هذا البحث تم استخدام لغة ++C واتضح أنها تتسم بكفاءة عند الاستخدام بما يعزز تطور نماذج البرمجة الكيانية وخصوصا في إدارة التخصيص الاحتمالي لقاعدة بيانات مكتبة, كما اظهر البحث إن تنفيذ البرمجة الكيانية يعد مرنا وقل كلفة عند التطبيق.

ABSTRACT

Computerized information system has recently become widely available. Indeed, it has become an important instrument that can be applied in different aspects of life. One of the important applications of computer systems is that adopted by commercial organization as well as by small business. Hence, we can say that no management can successfully operate in any business without the use of computerized information system.

This paper aims to design and implement probabilistic assignment management system for a library database commercial library. Most applications of OOP use specialized object oriented languages, but this paper shows that ++C can be used effectively to develop object oriented models of typical library management, and implement OOP is more flexible and reliable and less costing.

Introduction:-

Computerized information system has received much attention in recent years and it became widely available indeed; it has become an important instrument that can be applied in different aspects of life.

One of the important applications of computer systems is that adopted by commercial organization as well as by small business .That ,we may admit that no management can successfully operate in any business without the use of computerized information system .

The success of any organization is tightly coupled to its ability to store and manage information, Database storage systems provide a critical part of an organization's network infrastructure.(1)

Different approaches and technologies can be used to achieve different database requirements. The use of one approach or another depends directly on application requirements.

However technology by itself doesn't do anything; value is achieved when technology is utilized as a strategic tool to support organization information system.

One of these results is to maximize the profit of the establishment, and at the same time to keep hold its position in the face of other competitive forces.

The competitive advantage could be obtained by designing strategic information system. Objected Oriented Programming (OOP) is one of the philosophies that are being used in designing and implementing strategic information systems.

This paper is devoted to the study of the probabilistic assignment model. To motivate this study, we first discuss the deterministic assignment model. Think of allocating indivisible objects to agents who have strict preferences over these objects. Each agent can consume one and only one object. The idea is to discuss general methods (hence forth rules) of assigning objects to agents, as a function of agents' preferences. As each object may be given to at most one agent, there may be conflicts of interests.

Suppose a rule recommends an assignment for a particular group of Agents, objects, and preferences (henceforth referred to as a "library").

To illustrate the concept of probabilistic assignment model, we can consider a bookshop (library) which sells both books and video – tapes. The library provides different titles and prices of publishing. We can also create two classes one for storing the years published books and another for storing the playing time of tape. Imagine that a subgroup of the agents "reapplies" the rule to the library consisting of themselves, the objects they were assigned, and their induced preferences over these objects. The assignment recommended by the rule for this "reduced" library assigns each agent in the subgroup to the object it was initially assigned.

The probabilistic assignment model is a simple generalization of this deterministic environment, motivated by fairness considerations. For example, a common requirement in Models of fair allocation is that rule should not discriminate between agents with similar characteristics. This is the well-known property of **equal treatment of equals**. Clearly, as objects are indivisible, this property cannot generally be satisfied. By allowing rules to randomize over assignments; however, we can ensure that agents with similar characteristics.(2)

A program that allows an individual to have a set of library display contents is a library management program.

Object oriented programming is an approach to computer programming developed over the past twenty years. It is based on the idea that programs should represent the interaction between abstract representations of real objects. This differs from procedural programming, which is corresponded to a step-by –step list of computations without any necessary correspondence to natural processes.

Object oriented approach which has several advantages for this kind of implementation, library management system can be described in much more natural form by using an (OOP) approach and the different object may interact with each other in different forms offering great flexibility in the implementation.

One of the most useful features of (OOP) is the ease with which major change in models can be introduced with a minimum recording .Thus, an (OOP) language (C++) can model the complex library management system very flexibly and simply, so this

paper presents a brief overview of (OOP) in library management system, then we will consider the system components and the structure of object-oriented model.

A Probabilistic Assignment Model

Let N be a set of potential agents such that $|N| \geq 3$. The set of nonempty, finite subsets of N is denoted \mathcal{N} , with typical element N . Let X be a set of potential objects such that $|X| \geq 3$. The set of nonempty finite subsets of X is denoted \mathcal{X} , with typical element X .

Let $R \in \mathcal{X}$. A preference R over X is a binary relation over X which is complete, transitive, and anti-symmetric. Thus, preferences are strict. Let the set of preferences over X be denoted by $R(X)$. A library is consisting of a pair $(N, X) \in N \times X$ such that $|N| = |X|$, and a preference profile $R \in R(X)N$. The set of libraries is denoted by E .

An assignment for (N, X) is a connection between N and X . A typical assignment will be written μ . The set of all assignments for (N, X) is denoted by $A(N, X)$. A probabilistic assignment for (N, X) is a probability distribution over $A(N, X)$. The set of probabilistic assignments for (N, X) is denoted by $A(N, X)$. Note that risk preferences of agents are not Observed; so that all assignment decisions must be made by using only ordinal information.

A rule r is a correspondence which associates with each library (N, X, R) a non empty subset of $\Delta(A(N, X))$. (3)

Any assignment can be identified with a permutation matrix. Thus, by the theorem of Birkhoff and von Neumann, any probabilistic assignment induces a bi stochastic matrix, and conversely reduced model in which assignments are specified as the set of bi stochastic matrices. For our purposes, we are not justified in working directly with the set of bi stochastic matrices.

The reason is that the mapping which takes the set of probabilistic assignments into the set of bi stochastic matrices is not one-to-one. Thus, many probabilistic assignments may induce the same a bi stochastic matrix. This is without the loss of generality when discussing concepts which relate only to the welfare levels of agents; but when discussing conditions which relate *directly to the structure* of assignments, it is not without the loss of generality.

Look at Object Oriented Programming

An approach provides away of modularizing programs by creating partitioned memory area for both data and functions that can be used as templated for creating copies of such modules on demand.

OOP treats data as a critical element in program development and dose not allow it to flow freely around the system .It ties data more closely to function that operates on it and protects it from accidental modification from outside functions.

OOP allows us to decompose a problem into a number of entities called objects and then builds data and functions around these entities. The data of an object can be accessed only by functions associated with that object. However, functions of one object can access the functions of other objects.

The entire set of data and code of an object can make a user define data type with the help of a class.

In fact, the central concept of OOP is the concept of a class, which also one of the most intuitive and appealing feature the view point of investment management.

A class describes both the structure of an object and a set of procedures of initializing and using it .Once a class has been defined, we can create any number of object belonging to that class with which they are created.

Classes thus a collection of objects of similar type. Thus, classes are user – defined data type and behave like the built-in type of programming language. Data encapsulation is the most striking of class, the data is not accessible to the outside world and functions, which are wrapped in the class can access it .However, we can say that the wrapping up of data and functions into a single unit (called class) known as encapsulation, also OOP classes use the concept of data abstractions that refers to act of representing essential features, without including the background or explanation.(4)

This paper presents a brief overview of OOP in the context of library management.

The Structure of Object – Oriented Model

In order to appreciate the potential value of object-oriented modeling, it helps look at the relationship between models and the library database management they are intended to represent .We put the following set of definitions:

- 1- We design the algorithm of library database management system by missing OOP philosophy, since this philosophy has many features that make design strategy easier and reliable than other design philosophy.
- 2- We decide to use C++ as a tool to implement the algorithms designed by OOP philosophy because this language is the most popular one, and it cooperates with all features of the OOP designed methodology.
- 3- The model considers a library which sells both books and video – tapes. The library provides the title publication and their prices, also it can show the published year. The following program in the appendix illustrates and shows how to create a class known as lib that stores the title and price of a publication. We can then create two derived classes (5) , one for storing the published year of the books and another for storing the playing time of tape. The following figure shows the form of inheritance that can be used for writing extensible program; the direction arrow indicates the direction of inheritance class hierarchy for library.

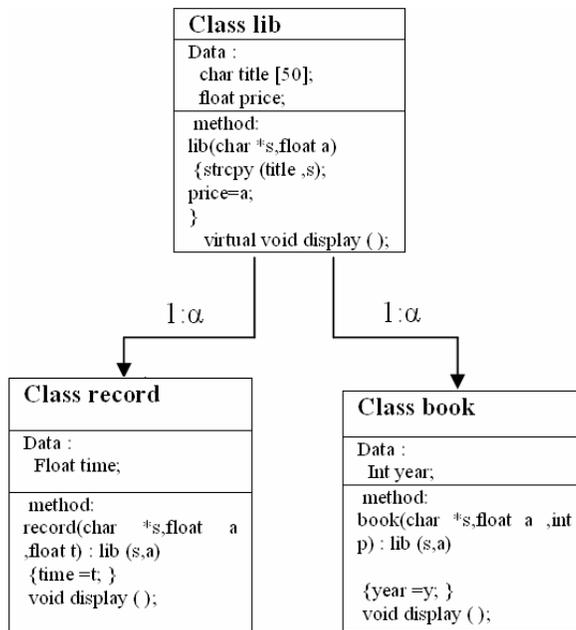


Figure (1): shows the class hierarchy for library.

The following program in the appendix illustrates and shows how to use the class variable for holding store details and how to construct this variable at run time by using dynamic initialization.

- 4- The technique which is used to create the derived classes (book and record) is called inheritance. It means that the relation between such classes that some of members declared in one class called the base class are also present in the second class called the derived class .The derived class inherits the members from the base class.
- 5- OOP simulation model consists of objects with complex internal dynamics which interact with each other. The rules of interaction are determined by properties of the object.
- 6- In this simulation model class, objects can be initialized dynamically too. That is the initial value of an object may be provided during run time the advantage of dynamic initialization is that we can provide initialization formats, by using constructors. This provides the flexibility of using different format of data at runtime depending upon the situation.
- 7- C++ provides a special member function called constructor which enables an object to initialize itself when it is created .This is known as automatic initialization of objects .The program uses parameterized constructors, the parameter values to these constructors are provided at run time, the user can pass the initial values as arguments to the constructor function when an object is declared .The argument of these constructors is the pointers to the title of publication and the price of its. (6)
- 8- In this program, we used one of the crucial features of OOP which is called polymorphism, it simplified means “one name, multiple forms ”. In C++, polymorphism is implemented by means of virtual functions and dynamic binding. When a virtual function is called via a pointer, the class of the object pointed to determines which function definition will be used. Virtual functions implement polymorphism, whereby objects belonging to different classes can respond to the same message in different ways.

A function [**display()**] is used in all classes to display the class contents. since the function **display()** has been virtually declared in lib, the base class.

- 9- An OOP simulation model consists of pointers. The C++ language provides a very powerful way to process data indirectly. That is, to work with values by remembering where they are in memory instead of what their names are.

In the (**main** program), we create a heterogeneous list of pointers of type **lib** as below

```
Lib *list[2]= {&book1,&record1};
```

The base pointer **list [0]** and list[1] are initialized with the addresses of objects book1 and record1 respectively.

The program structure is shown in the following figure (2):

a- The base class:

Class lib
Data : char title [50]; float price;

method: lib(char *s,float a) {strcpy (title ,s); price=a; } virtual void display ();
--

(a)

Base class name = lib

Title of publication (books, magazines, papers----etc.) = title

Price of publication = price

b- The derived class 1:

Class book
Data : Int year;
method: book(char *s,float a ,int p) : lib (s,a) {year =y; } void display ();

(b)

Derived class name = book

The published year = year

c- The derived class2 :

Class record
Data : Float time;
method: record(char *s,float a ,float t) : lib (s,a) {time =t; } void display ();

(c)

Derived class name = record

Playing time = time

Figure (2: a, b, c): program structure

The following program illustrates how to use the class variable for holding library details and how to construct these variables at run time using dynamic initialization.

Summary:-

In this paper, the OOP concept is applied to design and implement probabilistic assignment library database management system for commercial library.

This paper introduces a notion for the probabilistic assignment model. We show that the axioms *equal treatment of equals* and *probabilistic* characterize the uniform rule, which is the rule that randomizes uniformly over all possible assignments.

An OOP offers many advantages for developers of a probabilistic assignment library database management system models. We concluded from this paper that OOP approach to design a probabilistic assignment library database management system introduces some benefits that other approaches can introduce them. Some of these benefits are reusability that enable us to use the previously written functions in our paper.

Another feature is flexibility that means we can change some functions without changing the program structure due to these features.

Whether OOP languages have reached the point where they offer the power and efficiency to support large –scale-modeling projects is not yet known.

REFERENCES

- [1] René Hexel, 2010, Object Oriented Programming in Objective-C, School of Information and Communication Technology Griffith University.
- [2] Bogomolnaia, A., Moulin, H., 2001. A new solution to the random assignment problem. *Journal of Economic Theory* 100, 295–328.
- [3] Bogomolnaia, A., Moulin, H., 2002. A simple random assignment problem with a unique solution. *Journal of Economic Theory* 19, 623–635.
- [4] Bulgar sami(1997),”object oriented programming with C++” ,john wally &sons ln.
- [5] Coplien,j.(1992),”advanced C++ .programming styles and idioms. reading, mass. : Addison -wisely publishing co.
- [6] Christopher P. Chambers*(2004),Consistency in the probabilistic assignment model, *Journal of Mathematical Economics* 40 (2004) 953–962
- [7] Danilij cougar (1975)”introduction to computer based information system” , John Wally &sons in.
- [8] Rick hesse ,Assignment models and degenerate reports, *journal of mathematical economics* 2004 .
- [9] Radwan tahboub ,dr mahmoud al-sahib ,security in database web access ,*journal of al-rafidian university college for sciences* ,no.6 fourth year 1422-2001.
- [10] Thomson, W., 2003. Consistent Allocation Rules. Manuscript. University of Rochester.

Appendix I

The program:

```
# include <iostream.h>
# include <string.h>
Class lib
{
    Protected:
    Char title[50];
    Float price;
    Public:
    lib(char *s,float a)
    {strcpy (title ,s);
    price=a;
    }
    virtual void display ();
};
class book : public lib
int year;
public:
book(char *s,float a ,int p) : lib (s,a)
{year =y; }
void display ();
};
class record : public lib
float time ;
public :
record(char *s, float a ,float t) : lib (s,a)
{time =t; }
void display ();
};
void book:: display ()
{
Cout <<"\n" Titli : " << title ;
Cout <<"\n" published year: " << year ;
Cout <<"\n price: " <<price;
}
void record:: display ()
{
```

```
cout <<"\n" Titli : " << title ;
cout <<"\n" play time: " << time <<
"mins" ;
cout <<"\n price: " <<price;
}
main ()
{
char *title;
float price,time;
int year ;
cout << " \n ENTER BOOK DETAILS
\n";
cout << " TITLE :";
cin>>title ;
cout << " PRICE :";
cin>>price ;
cout << " year :";
cin>>year ;
book book1( title ,price , year);
cout << " \n ENTER TAPE DETAILS
\n";
cout << " TITLE :";
cin>>title ;
cout << " PRICE :";
cin>>price ;
cout << " PLAY TIME (mins):";
cin>>time ;
lib *list [2];
list [0]= &book1;
list [1]=& record ;
cout << " LIB DETAILS " ;
cout << "---- book ----" ;
list[0] → display (); // display book
details
cout << "---- record ----" ;
list[1] → display (); // display record
details
} (7)
```

Appendix II

- (1) Radwan tahboub ,dr mahmoud al-sahib, security in database web access ,journal of al-rafidian university college for sciences ,no.6 fourth year 1422-2001.p86 .
- (2) a-Bogomolnaia, A., Moulin, H., 2002. A simple random assignment problem with a unique solution. Journal of EconomicTheory 19, p 625. and
b- Rick hesse ,Assignment models and degenerate reports, journal of mathematical economics 2004 p956.
- (3) Rick hesse ,Assignment models and degenerate reports, journal of mathematical economics 2004 p961

- (4) Bulgar sami(1997),”object oriented programming with C++” ,john wally &sons ln.p66.
- (5) Bulgar sami(1997),”object oriented programming with C++” ,john wally &sons ln.p 136
- (6) A- Coplien,j.(1992),”advanced c++ ,programming styles and idioms. reading, mass. : Addison -wisely publishing co p123

Also:

- b- Bulgar sami(1997),”object oriented programming with C++” ,john wally &sons ln.pp 97-99.
- (7) The output of program would be:
 - Enter book details
 - Title : object oriented
 - Price : 100
 - Publishing year : 2000
 - Enter record details
 - Title : computer network
 - Price : 80
 - Play time (mins): 40
 - Lib details
 - book ----
 - Title : object oriented
 - Price : 100
 - Publishing year : 2000

 - record ----
 - Title : computer network
 - Price : 80
 - Play time (mins): 40