International Journal of Soft Computing 2 (1): 1-4, 2007

Information Systems: Object Oriented Development

Haramis, G.E., Th.A. Fotiadis and [1]John Mylonakis
University of Macedonia, Economic and Social Sciences,
156 Egnatia, Thessalonica, 54006, Greece
[1]Hellenic Open University (Tutor), 10 Nikiforou str.,
Glyfada, Athens, 166 75, Greece

Abstract: The aspects of Information System's development within the frame of object oriented technology are presented in this study. This methodology enables systems analysts and programmers to develop easily understood and ready-to-use software. More specifically, a procedural approach is proposed for object oriented analysis/design and programming.

Key words: Information systems, object oriented analysis, design and programming

INTRODUCTION

An object is defined by the abstract of a particular topic. The object "Daily List" is defined by data referring to any type of list, like, for instance, the data Title, Report, Page Number, etc.[1]. According to conventional methodology there would be a program created (usually in COBOL) with a view to reading various data from some files and developing this Daily List. This Traditional approach to the problem accepts the program and its items/file fields as two different entities. The Object-oriented approach, by contrast, defines an object called Daily List and according to this approach the program and the data are included in an object which creates the Daily List.

Object-oriented Development is a recent methodology that appeared in the mid-1990s Hoffer and Valacich[2]. This methodology enables systems analysts and programmers to develop easily understood and ready-to-use software[3]. The Object-oriented technique can be perceived as the conjunction of the data-oriented technique and the procedure or process-oriented one.

In the object-oriented technique the data and the processes are embedded in objects[4]. Thus, an object contains the data and the processes that can use or update these data. Only the processes assigned to certain data in an object can use/update this object. Different studies and object types interact with one another by sending messages, which command the execution of specific processes within an object.

OBJECT-ORIENTED SYSTEMS ANALYSIS AND DESIGN

According to the conventional methodology of developing information systems the distinction between entities and relationships that are significant to the users, is a usual technique. Object-oriented methodology preserves this distinction throughout the systems design and implementation phases. This has two advantages:

- it makes software intelligible to Business Executives, who can easily understand anything related to business processes and
- the objects can be used afresh in implementing similar enterprise information systems of the same business environment.

This capability to use a basic object and merely add coding using a computer language for the extras of a new system saves time in systems development as well as in maintenance.

Objects represent re-usable modules. As an object contains its own operations and its own data, it can function as a self-contained module that is independent from other modules. As already said an object interacts with another only through transmitting and receiving certain messages. An object will perform an operation upon receiving a message that will pass some necessary parameters to it, which will cause the object to execute its purpose.

Corresponding Author: Haramis, G.E., University of Macedonia, Economic and Social Sciences, 156 Egnatia, Thessalonica, 54006, Greece

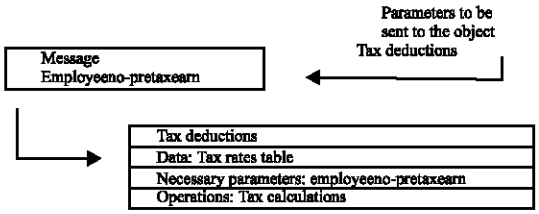

Fig. 1: An example of sending a message to an object

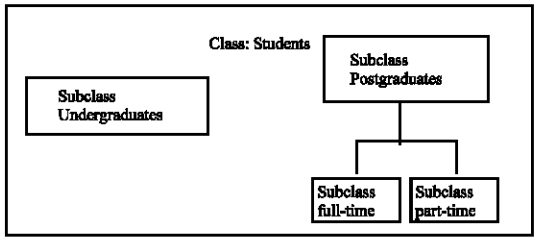

Fig. 2: An example of a class and subclasses

The Fig. 1 is a relevant example; it refers to estimating the income tax of a Company's employees. A message with each employee's number (EMPLOYEENO) and the amount of his pre-tax earnings (PRETAXEARN) is the argument sent to the object Tax Deductions and the object responds by calculating the total tax.

The messages are similar to calling a sub-routine in traditional methodology. Object development should not occur from scratch[5], as is unfortunately the practice for the time being; existing objects should be inspected in search of components that could be used in developing a new object. Only when there are no such components, should objects be developed from scratch. Therefore, the new object and its components will be henceforth available for creating a new object.

The above study for object development can be called an information factory approach, in that it resembles the way in which many new products manufactured in an industry are based on existent designs and processes. As objects are independent from each other, they can be tested separately. Following a successful test, these objects are ready to be connected with other objects and form an information system.

Once an object has been made ready, it can be used, as mentioned earlier, in the development of similar objects that will have identical behaviour and features. Objects that come from or are related to one another form a class of objects. Each class includes special guidelines or methods, which are unique to the class in question.

Analysts can develop subclasses of objects that have all the components of their parent class, but whose features are unique to them. For example, the class

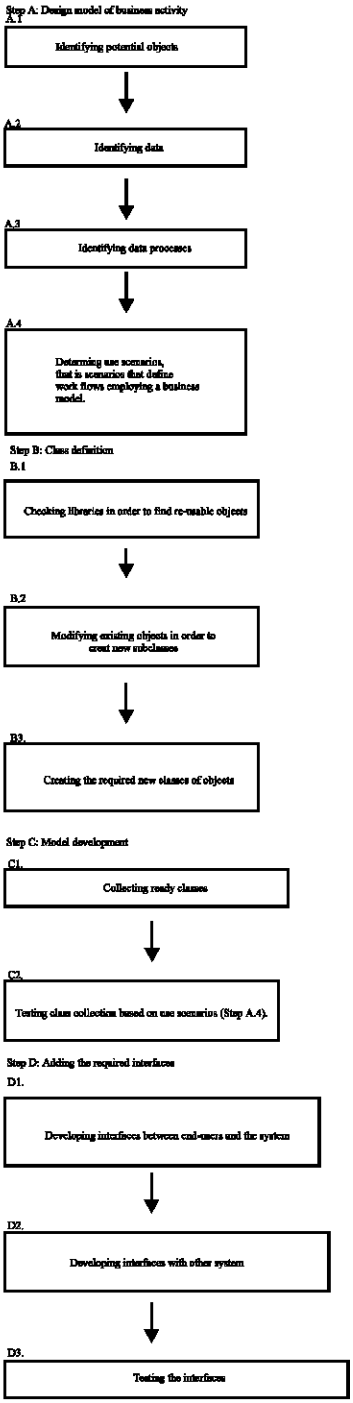

Fig. 3: A Diagram showing the Object-oriented Development of Information Systems

students can be divided into subclasses; the subclass of Undergraduates and the subclass of Postgraduates Fig. 2. Each of these subclasses, belonging to the class of students, can be said to inherit the features of the

students' class, for it has common attributes with it, i.e. Full name, Year of studies, etc. and also to have its own special features.

Afterwards, each subclass can be further divided into further subclasses, like the postgraduates' subclass, for instance, which can be divided into Full-time and Part-time ones, each new subclass again inheriting the features of students and also having its own special features. Therefore, at any rate a subclass inherits all the features and capabilities of the (sub) class above it. In object-orientated technology terms, this is called Inheritance.

Hence, all objects are defined by class and inheritance. To come back to the issue of re-using objects and in relation to the foregoing facts, it could be stressed that the object oriented systems development becomes more effective as the number of libraries with re-usable objects increases. Thanks to such libraries, coding update becomes far easier, in that, should the parent class design need to be modified, all class members are automatically updated so as to respond to that modification.

Object-oriented Analysis defines systems requirements by creating systems models using related objects and related classes. Object-oriented Design is the method of defining objects for an object-oriented program at the logical and physical level. Object-oriented analysis, object-oriented design and the object-oriented process of developing information systems are illustrated in the following diagram Fig. 3, which is developed in four basic steps:

The first step (Design Model of Business Activity) looks like the traditional methodology of systems analysis and design. The second step results in creating new classes of objects. The third step refers to developing an operable model. The fourth and last one defines the required interfaces that need to be added in the process.

OBJECT-ORIENTED PROGRAMMING

Developing as well as maintaining programs is known to be time-consuming and very costly. According to traditional methodology, programmers have invariably started to develop a program from scratch, even in study they have known that there is a/there are similar program(s); this has been happening because it is much easier for them to study the requirements of a new program and develop it rather than investigate what an existing program does (albeit a fully-documented one) and modify it so as to meet these new requirements.

Object-oriented Programming, by contrast, is a program development methodology aimed at producing program modules that can be re-used by other programs. This re-usability is the main feature of object-oriented programming, whereby every module configuration in a program is an object containing all the data and the instructions that act upon them.

The philosophy is the same as when a manager asks one of his subordinates to prepare a sales list[6] the subordinate knows from where to take the necessary data and how to prepare the list and as a result, the manager is certain that whenever he asks for a sales list, his subordinate will produce a list comprising particular data presented in a particular way.

As a general rule, an object contains a programming code that combines (as already said) data and instructions related to the operations that will be performed on these data. The code describes all the attributes of the object and all the operations that the object may be requested to perform. When an object-related operation is being performed, a message is sent to the object; this message merely defines the operation that is to be performed; how it will be performed is already within the instructions, which are part of the object.

In the example of the class students, every object first has all the data related to a student, e.g. matriculation no, full name, etc. and secondly, operations that can be performed on these data, like, for example, changing the year of studies, entering a mark, etc.

Object-oriented programming employs objects[7] as logical parts of a program instead of designing a logical diagram or developing an algorithm and then coding either in a language of logical activities of a diagram or algorithm.

Small Talk, which was developed by Xerox, has been the first object-oriented programming language. C++ ensued and today several programming languages enable programs to be developed in an object-oriented fashion technique. After all, as it has been already made clear, object-orientated programming principles can be utilised by the existing conventional languages, too.

CONCLUSION

The object-oriented technique will dramatically change the methodology of developing information systems in that, once it is adopted, the significant advantages of re-usability, speed and scaled-down maintenance are achieved.

Nowadays, object-oriented technology is popular with software and hardware companies. Analysts and programmers though are still cautious; as a matter of fact, this technology has been widely-known in the field of programming, but in the domain of analysis/design it has appeared rather of late. Besides, an issue that has yet to be resolved is that considerable time is needed for analysers and programmers to be trained anew.

One thing is certain; it is imperative that information systems be developed faster and maintained at a lower cost. The next decade will show whether it will be the

object-oriented technology or the automatic systems development/maintenance with the use of CASE tools that will prevail.

REFERENCES

1. Wu, M. and S.Y. Wu, Systems Analysis and Design, West Publishing Co.
2. Hoffer, H., J., Valacich and J. George, Modern Systems Analysis and Design, The Benjamin/ Cummings Publishing Co., Inc.
3. Beynolds, G., Information systems for managers, West Publishing Co.
4. Edwards, P., Systems Analysis and Design, Mitchell McGraw-Hill.
5. Whitten, J., L. Bentley, V. Barlow, Systems Analysis and Design Methods, Irwin.
6. Hicks, H.O., Jr., Management Information Systems: A User Perspective, West Publishing Co.
7. Alter, S., Information Systems: A Management Perspective, The Benjamin/Cummings Publishing Co., Inc.