OBJECTive teaching: an experience in teaching object-oriented technology

S. Ashok
Information Communication Institute of Singapore, 1 Hillcrest Road, #08-00, Tower Block 2, S(1128), Republic of Singapore

ABSTRACT

The Object Oriented (OO) technology has a lot of benefits to offer in the field of software development. The technology is promising in terms of increased productivity of developers, software reusability and maintainability. These benefits become significant in large and complex software projects. Communication Software projects being inherently large and complex, could reap the benefits of this technology substantially. At Information Communication Institute of Singapore (ICIS), we recognized this potential and introduced the OO approach in our Master's programme curriculum in 1993.

Communication Software Systems are event driven in nature and exhibit dynamic behavior. Also, the very vital activity of identifying objects is not straightforward since problems in this domain cannot be easily related to the real world. Not many methodologies addressed these aspects in desired depth. As a result, one of our greatest difficulties was to conceive a suitable methodology for Communications Software development and teach it to the students.

In the course of our teaching, we also had to face the challenge of moving from the structured approach of software development to the OO approach. It is certainly a paradigm shift and requires a new way of thinking and approaching a problem. One of our challenges was in driving home the OO perceptions and techniques into the minds of students particularly groomed in the functional approach of Communication Software development. Due to the abstract nature of the Communication Software domain, these challenges were compounded.

This paper shares our experiences in conception, preparation and delivery of an OO module in our curriculum. It describes the difficulties encountered in preparation of the module and how the problems were solved. This paper also discusses the various teaching techniques used in teaching the OO Concepts, an
OO Methodology and OO Programming as applied to the Communication Software development process.

1 INTRODUCTION

Information Communication Institute of Singapore (ICIS), a collaboration between AT&T and National Computer Board of Singapore[1], is one of the premiere institutes of Singapore, offering Master Degree courses in Communication Software Management. The institute produces specialists in the area of Communications Software, by imparting in-depth knowledge in three areas, namely Communications Technology, Software Development Process Technology and Platform Technology[2].

About 33% of the classroom hours are spent in teaching modules in the software development process technology area. The primary methodology taught is the Waterfall Model using Yourdon's Structured Approach.

At ICIS, we periodically review the curriculum and strive to incorporate the latest technology. In our efforts to upgrade our software development process technology modules, we perceived a need to introduce the Object Oriented (OO) Technology as a part of the curriculum, in 1993. As claimed by many academicians and practitioners in this area, the OO technology promises to be beneficial in terms of increased productivity of developers, reusability and maintainability of software. This is especially significant in large and complex software projects. We recognized that Communications Software being of this nature, could reap the benefits of this technology.

This paper shares our experiences in the introduction of an OO module in our curriculum. The paper describes the difficulties encountered as well as how the problems were solved in the three phases of the development of this module namely, conception, preparation and delivery. Three areas namely the OO concepts, OO methodology and OO programming as applied to the Communication Software Development process are dealt with. The various teaching techniques employed in each area are also elicited. Finally, the paper concludes by highlighting the success and the failure of the introduction and teaching techniques employed in this module.

2 CONCEPTION OF THE MODULE

Gearing towards the Nation's Vision of IT2000, the National Computer Board launched its National Information Infrastructure (NII) project in 1992[3]. It was identified that the OO technology would play a vital role in realizing the NII. It can be easily visualized that the backbone of the NII would be a complex network, capable of supporting distributed processing and multi media traffic. Being an active participant in the NII project, we envisaged that building such a complex software system using the structured software development approach, would be an unwieldy task. Considering the overwhelming benefits of the OO technology, it was rather compelling to expose our students to this technology.
With this strong motivation, we decided to introduce a short module in our curriculum.

The objective of this module is to give the students an insight into the basics of OO, a suitable methodology to build Communications Software Systems and an introduction to OO programming. The module was planned to be of 14 class room hours duration and was to be offered in the third quarter of the 1 year course. The module was titled "Introduction to Object-Oriented Analysis, Design and Programming". The module plan that was laid out is shown in Table 1.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Topic</th>
<th>Duration(hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to OO Concepts</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>An OO methodology</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Other OO methods</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>OO Programming</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td><strong>14</strong></td>
</tr>
</tbody>
</table>

### 3 PREPARATION AND DELIVERY OF THE MODULE

The module was developed by strictly following the ICIS Quality Standards. Keeping in mind the background of the students, the module had to be developed with great care so as not to confuse the students about the perception of the software development process.

This section will describe the difficulties encountered during the preparation of the units 1, 2 and 4 (refer Table 1) and how the problems were tackled. This section will also deal with the teaching techniques that were used to deliver these units.

### 3.1 Unit 1 - Introduction to OO concepts

**Objective**

The objective of this unit is to introduce some of the basic concepts and to highlight the benefits of this technology over the structured approach.

In terms of the concepts, this unit highlights the four basic concepts namely, Abstraction, Encapsulation, Inheritance and Polymorphism. The difference between a class and an object (an instance of a class) and the relationships between the objects such as 'using', "aggregation" etc. are also brought about.
**Preparation**

Difficulties in the development of this unit were very minimal. The OO concepts being well established, substantial literature could be found in this area.

Illustrating the concepts with good and clear examples is very crucial since the foundation for the understanding of this new technology is laid in this unit. Looking for illustrative examples, relevant to the communications domain is not an easy task. The domain being so abstract, examples from this domain may not be well understood by beginners in the field of OO. We therefore chose examples from the real world to explain the OO concepts.

Some of the concepts being subtly different (and application context dependent), could be very confusing if clear examples are not chosen. One such example is that of the concept of an object (instance of a class) Vs a subclass. In Figure 1, Mercedes, Nissan, Toyota etc. will be considered as a subclass of the class CAR in a car rental application. Whereas, in a car statistical report application, they will be considered as objects of the class CAR.

![Diagram](image)

**Figure 1**

**Delivery**

The most difficult concept to drive home is the concept of polymorphism. In our opinion, this concept is relevant most often during the implementation stages. And hence can best be explained with the help of an implementation language. Nonetheless, it is one of the fundamental concepts that one has to understand when using the OO Technology.

The benefits of OO in terms of it being the most natural way of approaching the problem was highlighted during the course of teaching this unit. This is done by way of enacting a story of a five year old child narrating the story of his visit to the airport. The objects identified by the child in the airport (e.g.: airplane, runway, coaches etc.) and the attributes (e.g.: attributes of airplane are engine, wings, wheels etc.) and services (e.g.: services of the airplane are taxi, fly, takeoff, land etc.) of each of the objects recognized by the child are all illustrative of the concept of OO. Reflecting such similarities and relating the problems to the real life problems were found to be very effective in reinforcing the different approach to problem solving using OO technology.

Also, during the delivery of this unit, we discovered that the discussion on the benefits of OO Technology and the advantages of this technology over the structured methods can best be appreciated after the OO concepts are taught.
In our opinion, the effectiveness of teaching this unit is very vital. Putting in a lot of effort in teaching this unit definitely pays off when it comes to teaching the subsequent units of this module.

3.2 Unit 2 - An OO Methodology

Objective
The objective of this unit is to introduce a methodology for developing software in the communications domain. The methodology taught primarily highlights the activities and stages of the software development using the OO technology. It takes the students through the Analysis and Design phases using an integrated case study of a communication problem.

Preparation
Most of the available methodologies focus on OO analysis and design issues pertaining to problems in the commercial domain. The analysis in most cases starts with identifying the objects based on the real world objects. In fact, very few methods have a systematic approach to the identification of objects. This made it difficult for us to choose a method applicable to problems in the communications domain. The problems in this domain being abstract in nature, required a method that can cater to the dynamic nature of the problems and has a concrete process for identification of meaningful objects.

We evaluated several methodologies like OMT by Rumbaugh[4], CRC method by Wirfs-Brock, Booch method, Shlaer and Mellor method, Coad and Yourdon method and OBA by Adele Goldberg[5]. None of these methods were found fully suitable for problems in our domain. We therefore formulated our own methodology. This methodology uses a modified version of the OBA method for the object identification activity and then uses the modified version of the OMT method for the rest of the analysis activities. As for the design activities, we enhanced the method suggested by the OMT method.

We chose the CCITT’s X.224 Recommendation [6] (Transport Layer Class 0 - TP0 Protocol) to try out our methodology. TP0 is the fourth layer protocol of the OSI Communication stack. This layer is responsible for providing end-to-end connectivity between systems, and hides the details of the network from the users of this protocol. We found the experiment very encouraging. In fact, this example was illustrative of almost all the concepts of OO Technology including the reusability.

By going through the above exercise, we also realized that the functional modeling step of OMT, was very tedious and it did not add much value to the analysis process. This in our opinion is because the functional approach and the OO approach are fundamentally diverse and hence do not blend well. Hence, we modified the method excluding the functional modeling.

The unit was developed using the TP0 case study, taking through the analysis and design steps constantly highlighting the various activities in each step.
Delivery
Using the TPO case study demanded extra sessions of lessons, where in-depth understanding of TPO was imparted. This made us believe that taking the OO approach entails one to be well versed with the domain before attempting to analyze and design a system. The familiarity with the terminology of the domain is very important because the objects that emerge will have to be closely related to the nature of the problem. Choosing the right names for the objects reflects a good design.

3.3 Unit 4 - Programming

Objective
The objective of this unit is to give the students an exposure to the OO Programming using an implementation language. This unit does not cover all the features of the language taught. Rather it highlights the implementation of the OO concepts taught to them in this module.

Preparation
We picked the language C++ as the media to teach OO Programming due to the following reasons:

I. The language could illustrate the features of OO Programming effectively
II. The language is quite popularly used in the industry
III. The compiler for C++ is readily available in the UNIX platform
IV. Since the students are already familiar with the language C, learning the syntax of C++ will not be very difficult

The preparation of this module was not very difficult. The unit illustrated the implementation of the four most basic concept of OO technology namely Abstraction, Encapsulation, Inheritance and Polymorphism. It also illustrated the implementation of aggregation and using relationships.

Delivery
The most effective way of teaching this unit was to relate the new concepts to the concepts that the students are familiar with in the language C. By eliciting the fact that most procedural languages also attempted (though not completely successful) to achieve the concepts in OO Technology, it was found easier to hold the attention of the students. For example, the concepts of a class is an extension of the structure in the C language and encapsulation in C is attempted by way of static data types.

The concept and benefits of Polymorphism and Inheritance could easily be explained via concrete examples in this unit.
4 ASSESSMENT

The students were assessed by homework and class exercises in the OO concepts as well as OO Programming. As for the assessment on the methodology, the students were given a group (4-5 persons) project, because we believe that modeling an OO system can best be done in a group. The students were additionally assessed by their ability to participate effectively in the class discussions.

It is also observed that some students who do very well in the other software engineering modules do not do as well in this module. The reason being that starting from the time the students enter ICIS, they are grilled to have a “structured” mind-set. The software development process curriculum is primarily based on the waterfall model using the Yourdon’s structured approach. By the end of the second quarter, the students become hard-core procedural software developers. It requires them to break free from the habit of thinking procedural and tune their perception to the entities or objects that constitute a system[7, 8].

5 CONCLUSION

In our experience, we have found that teaching Object Oriented Technology is a real challenge. Especially to individuals who have a very strong background in the structured approach to software development. The most difficult thing to teach is to make them get rid of the old ways of approaching the problem from the procedural perspective and look at it in a more natural way. The focus has to be shifted from thinking top-down to bottom-up.

A strong base in the OO concepts is very vital in laying a good foundation to the understanding of the new paradigm. Clear and illustrative examples are extremely important as they become the vehicle to driving home the concepts. OO Analysis and Design can best be learned by practice and in a group. Therefore, giving students exercises in modeling OO systems becomes crucial.

Also, a sound knowledge of the domain is very essential to building a good system. Hence, case studies which are used to illustrate the methodology should not only be relevant but also should be well-known to the students.

Teaching the OO concepts and a methodology before teaching OO Programming forms a good foundation of the concepts of the technology and hence puts lesser emphasis on the nitty gritty details of the language syntax. This helps in using the OO languages in the right perspective, especially those that are hybrid languages like C++.

Relating new concepts to the already known concepts is found to be a very effective way of holding the students attention.
6  ACKNOWLEDGMENTS

The author wishes to thank Mr. Raymond Lim Long, Mr. Chan Ping Wah and Mr. Lim Say Beng for their continual encouragement. The author is also grateful to Mr. Lau Kai Cheong and Mr. K.R.S. Murthy for patiently reviewing this paper, and Mr. Sng Hock San for helping in editing.

7  REFERENCES


