

Teaching Object Oriented Analysis and Design to Undergraduates in CS

Sudharsan R Iyengar
Computer Science Department
Winona State University
siyengar@winona.edu

Abstract:

Object Oriented Technology permeates most application and system software development environment. It is also the choice of instruction in many undergraduate CS programs. This paper presents our method of introducing and teaching undergraduate CS majors the principles and techniques of Object Oriented Software Analysis and Design using UML and UP. The objective of the course (CS-371) is to develop a sense of quality industry-style analysis and design for large scale software using UML and UP. This paper presents a plan that has proved effective for preparing undergraduate junior/senior student population with OOA and OOD techniques.

Introduction:

Object-orientation is a technology based on the paradigm of objects and classes that match domain abstractions and lead to the development of universally understandable and usable framework for software engineering. Good modeling process and well understood implementation strategies provide a natural means of focusing on the problem model through methodical analysis, design, and development of the software. Pragmatic use of the paradigm is shown to greatly improve software quality.

Object-orientation can be utilized from the very start of software engineering process and provides a complete and comprehensive systemic approach to techniques, methods, processes.

OO paradigm is applicable over a range of domains which ranges from user-intensive applications to design and implementation of the systems that run beneath them and the distributed systems that connect them.

Organizations make great use of this model because OO provides for better software and analysis models, reusable asset, and responsive system that can be well customized.

OO frameworks are available that expand the capability of programming languages and thus open new horizons for creativity and use. These frameworks are platform independent, and promote universality of design and implementation.

Many object oriented methodologies are available in literature. Many of these were adopted and used widely. Some of them include the Booch method, Coad/Yourdon method, the FUSION method, Jacobson method, Rumbaugh method, etc.

Developing a model for an industrial-strength software system prior to its construction or renovation is as essential as having a blueprint for large building. Good models are essential for communication among project teams and to assure architectural soundness. As the complexity of systems increase, so does the importance of good modeling techniques. Among the many factors that contribute to the success of a project, having a rigorous modeling language standard is one essential factor.

The Object Management Group (OMG) is an international software industry consortium with two primary aims: a) promotion of the object-oriented approach to software engineering in general, and 2) development of command models and a common interface for the development and use of large-scale distributed applications (open distributed processing) using object-oriented methodology.

The Unified Modeling Language (UML) [1] is the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It simplifies the complex process of software design, making a "blueprint" for construction.

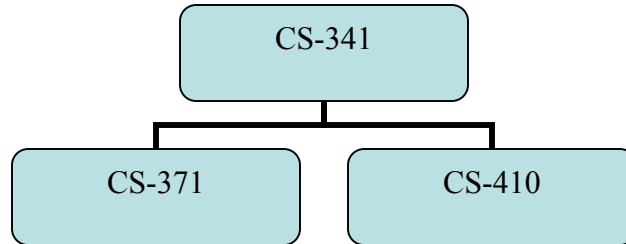
In short, OO paradigm, methodology, and technology are integral to industry needs, and this has become fundamental to any CS program. Problem solving and software development experience using these technology and tools, are invaluable to graduates seeking value in the workplace.

Demuth presents in [2] their experiences in bringing OOA and OOD to undergraduates using frameworks.

CS Program at WSU:

The Computer Science Department at Winona State University (WSU) offers programs and courses in computer science. These courses are designed to give students an understanding of and appreciation for both theoretical and applied computer science. Graduate of the program are prepared for careers in industry or further study at the graduate level. The department offers two majors – the Computer Science Major and the Computer Information Systems Major. The CS major imparts depth and breadth in computer science with aim of preparing them for CS industry and graduate programs in CS. The CIS major aims to prepare students with core CS background and relevant training for its application in industry that heavily use Information Systems.

Essential to both these majors are the core courses in Algorithm, Problem Solving and Data Structures, Discrete Mathematics and Calculus, and Fundamentals of Computer Architecture. Over and above both these majors require experience in Software Systems Engineering. The software engineering aspect of the program is provided through two courses – the Software Engineering (CS-410) and Object Oriented Analysis and Design (CS-371).



The twin courses: CS-371 and CS-410:

The course CS-371 is intended to emphasize the object oriented techniques and methodology through the use of appropriate tools. The course CS-410 deals with the broad range of software engineering principles and issues. Additionally, CS-371 has emphasis on oral communication and CS-410 has emphasis on written skills.

The catalog description for these courses are as follows:

CS-371: This course will cover fundamental topics in object oriented analysis, design, and development. An object oriented design methodology and tool will be introduced and used. The course will use an object oriented development environment/language. Advanced features of object oriented languages will be covered. Students will be required to investigate issues in object oriented systems and their implementation. CS

CS-410: The course deals with the current trends of software engineering principles and techniques for methodical construction of large, complex software-intensive systems. It follows the software life cycle from the requirement, specification, design, and testing phases, Topics include software process, project management, quality assurance, configuration management, formal specification techniques, design methodologies, testing, and validation techniques, and object oriented methodologies. Students are involved in a team project utilizing software engineering principles.

These courses are expected to be taken after successful completion of CS-341.

CS-341 (Advanced Data Structures) is the culminating course in the introductory sequence in Algorithms and Problem Solving with the emphasis on data structures, complexity of algorithms and algorithm design technique.

Course Syllabus and Activities for CS-371:

Given below are the handouts that describe the objectives and grading scale for CS-371.

CS-371

Object Oriented Design and Development

T R: 9:30 am- 11:00 am; WA 105

Objective:

The objective of the course is to study, understand, and appreciate the process of OOA, OOD, and OOP and develop skills with UML and UP.

Grading:

Exam 1
10 %

Text: UML and the Unified Process

Jim Arlow and Ila Neustadt; Addison Wesley

Exam 2
10 %

Lecture materials and references provided by the Instructor

Class Project
20 %

Office Hrs: MWF: 8:00 am – 9:00 am; 10:00 - 10:30 am

Special Project
20 %

TR: 8:00 am - 9:30 am; 11:00 am - 12:30 pm

Syllabus:

- UML and UP
- Elements of Requirements Identification
- Fundamentals of Analysis and Comprehension
- Design Patterns and Frameworks
- Development and Deployment
- Review

Attendance and Active
Participation
10%

Case Study
10 %

Final Exam
20 %

Total
100 %

This course may satisfy University Studies Oral Flag requirements following completion of the University's approval process.

GradingScale

Grade

>= 90%
A

>= 80%
B

>= 70%
C

>= 60%
D

< 60 %
F

Projects:

Students will experience and develop a sense of quality industry-style analysis and design for large scale software projects.

The primary means of education will be through the development of vision, collecting requirements, identifying and elaborating user requirements, team discussions, peer review, in-class presentations, use of an UML tool, and supplementary lectures. All these will occur throughout the semester.

Oral communication will be crucial for the dissemination of your vision, concept, analysis, pros and cons, and strategic design techniques.

A Class Project will be initiated and directed by the instructor. Different subgroups will address, develop, and implement different components through different phases of this project. Analysis and Design deliverables from one group will be used by others, to extend the project. This project will require extensive inter-group discussions for clarification and correction.

Projects:

As part of activities required toward the completion of the course requirements is the development of a large-scale analysis and design project that is spread over all the students in the class. This project requires inter group communication and inculcates in them a streamlined process of standardized input, workflow, and deliverables as advocated in the unified process (UP). Below is the guideline that the students adhere-to through the class project.

Guide for Class Project Object Oriented Design and Development.

The instructor will assist in forming groups of no more than 3 individuals.

The instructor will initiate an Application Software/System pertinent to an industry.

Groups will discuss and develop rationale for the project.

The class will collectively develop a vision and an UI model.

The class will also identify different users and components of the system.

The instructor will assign the identified users and components to groups for study.

Groups will study their tasks and present the identified use-cases and requirements to the class.

The instructor will integrate these and re-distribute the identified use-cases and requirements to groups for analysis.

Groups will analyze their tasks and present a completed analysis to the class.

The instructor will re-distribute the deliverables from analysis to groups for design.

Groups will use design strategies to complete, justify, and present their design for implementation and deployment.

Group members should retain all documentation and notes pertinent to the project discussions and presentations.

Additionally, as part of activities required toward the completion of the course is the development of the selection, investigation, research and the analysis and design of a specialized project in a domain that utilizes and exemplifies the use of OO techniques. Each group, undertake such special project and responsible for all stages of the development. Consequently, this requires intra- group communication and reinforces the

UP process. Below is the guideline that the students adhere-to through their special projects.

Guide for Special Group Project
Object Oriented Design and Development.

Form groups of no more than 3 individuals.

Identify an Application Software/System that you want to develop.

Propose and develop a business case and justification, for your project, and present this to the class.

Meet and brain-storm, perform research and field analysis, and start OOA.

Present feasibility arguments together with use-cases, and request for project approval by the class.

Complete a thorough design – ready for implementation. This should contain all the elements of a complete design.

Complete your analysis. Using design strategies complete the design for implementation and deployment.

Present your Design and determine resources for implementation – giving timelines with description of deliverables.

Discuss next steps for extending the application you have developed.

Group members should retain all documentation and notes pertinent to the project discussions and presentations.

Culminating their experiences with the process of UP is their research and presentation of a OO technology and/or tool. They are required to present this as a choice of development environment for the special project whose design (first stage) is nearing completion. Below is the guideline that the students adhere-to for their research and CASE tool presentation

Guide for Case Study (CASE tool)
Object Oriented Design and Development.

Identify a Technology, or CASE tool in the market that assists in OOOD.

Research, develop, and present to the class - addressing the following points:

History and Motivation

Unique and Specific Contributions to
Requirements engineering
Analysis Phase
Design phase
Development Phase
Transition Phase

Discuss its strengths and weakness.

Also, contrast this CASE tool with other similar systems and justify why we should invest in this technology.

Sequencing of the Project Activities and Presentations:

A class project is initiated by the instructor in the first week. The class works on the inception phase of this class project. They get their first experience in requirements and use-case engineering. It is at this juncture that groups are required to initiate their special project – which is around the 4th week of the semester.

The projects are undertaken adhering to the UP methodology.

Inception: Develop Vision, Identify and Finalize Requirements, Identify Use-cases
Elaboration: Detail Use-cases, Realize Use-cases, Design System
Construction: Implement, and test. (This phase not implemented in the course.)
Transition: Performance Test, Deploy, and Maintain. (This phase not implemented in the course.)

The deliverables from the inception of the class project is re-distributed to other groups. They then are involved in the analysis of these documents. This is their first experience in methodical OO analysis using deliverables from the previous phase. Simultaneously, they work on the requirements and use-case engineering (second experience in this inception) on their special project. As they complete the analysis of the class project – they are also

completing the inception of their special project. This phase is completed around the 8th week.

The deliverables from the analysis of the class project is re-distributed to other groups. They are then involved in developing the design from this analysis. This is their first experience in methodical OO Design from Analysis documents. Simultaneously, they work on the analysis of the deliverables from their inception (second experience in elaboration) of the special project. As the design of the class project is completed – they also work and complete the analysis of the special project. This phase is completed around the 12th week.

The students complete the design of the special project – their second experience in design. Simultaneously, they research and investigate a CASE tool appropriate for development. They complete this phase in the 14th week.

A tentative schedule followed by the students in Spring 2004 is given below:

CLASS Project:

Topic identification and initial discussion by instructor:	1/15/04
Vision and Scope; users identified:	1/22/04
Use-cases and requirements: (Class will have to critique and give the go ahead)	2/12/04
Presentation of Completed Analysis	3/11/04
Presentation of Design	4/1/04

SPECIAL Project

Identify application and justify topic to instructor:	1/29/04
Business case proposal: (Class will have to critique and give initial approval)	2/19/04
Presentation of Feasibility and Analysis: (Class will have to critique and give the go ahead)	3/18/04
Presentation of Completed Analysis	4/15/04
Presentation of Design and CASE tool	4/27-29

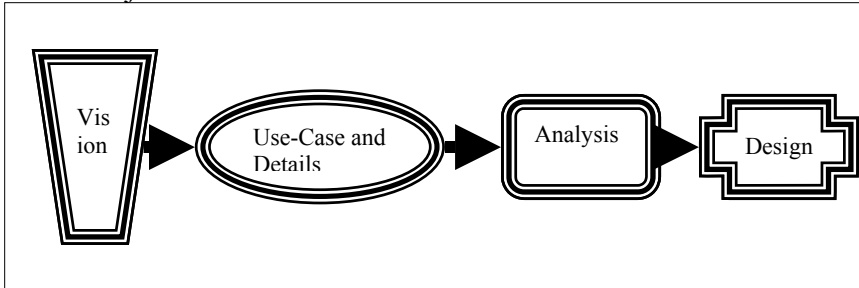
CASE Tool Study

Identify topic and justify topic to instructor:	4/13/04
Preliminary presentation to class: (Class will have to critique and approve the topic)	4/15/04
Final presentation to class:	4/27 - 4/29/04

The special projects are phased behind the steps in the class project, and consequently are the presentations. This workflow is illustrated in Figure 1. Students used Poseidon (from Gentleware.com) in Fall 2003 and will be using Rational Rose in Spring 2004. During the

Design phase the students are introduced and exposed to the various OO Design Patterns. They also study the need, use, and value of OO Frameworks. As they do the projects they see the immediate reuse of a developed framework in other projects.

Class Project:



Special Project

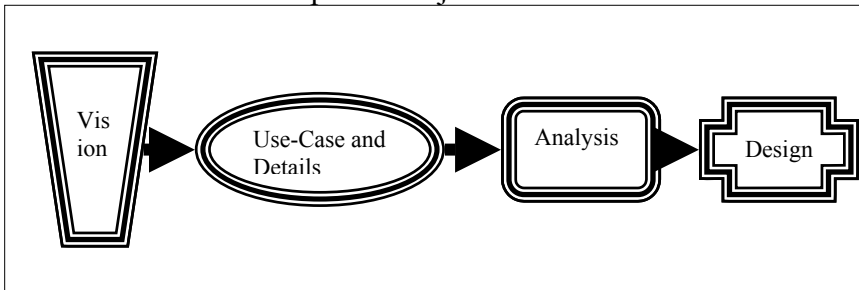


Figure 1: Phased Stages of Class Project and Special Project.

Projects:

A Class Project (HMS) was commissioned by the instructor. This project was discussed and worked by all members of the class. Special projects that were developed by student groups follow the same process and procedures. Groups worked on their special projects independently.

Case studies were also presented by the different groups. The topics were San Francisco, .NET, J2EE, and Delphi. These technology products were independently researched and investigated by student groups and presented for appropriateness for its adoption for the development of the special projects.

Given below – Figure 2. - are the vision statements of the different projects.

HMS - Vision

HMS (Hotel Management System) will be an integrated and unified system simplifying and assisting in customer management, customer service, and operational management of Hotels/Motels. It will be web-based system with the provision of hand-held devices for service personnel.

GCMS - Vision Statement

The Golf Course Management System (GCMS) is a system for managing golf courses.

The goal of the GCMS is to help organize golf course maintenance, and tee time and caddy scheduling operations. The GCMS also will have a web based interface for golfers to schedule tee times and caddies, and view their personal golf statistics on-line. The GCMS must have an easy to use web page for golfers to use for reserving tee times and caddies, and viewing their statistics. There must also be an easy interface for course employees to view both scheduled reservations and maintenance duties such as lawn mowing and watering, and sand trap maintenance.

WUB - Vision

Winona Used Bookstore system makes purchases, trading of books electronically via the internet. It provides all the functions like new customer registrations, returning books, customer lookups and searching of used books online.

This system will allow customers to purchase books for cheaper price or even trade books with other books online. Buying books from a bookstore can be expensive and even troublesome if you are looking for a specific book written by a specific author. This system will use a searching mechanism so customers can look up their books by the book's ISBN number or the title and the author of the book.

IntelliTracker - Vision

IntelliTracker (IT) is the newest all-in-one management package for the automobile sales industry. If you are an automobile dealer, this software will allow you to observe the latest statistics about the vehicles on your car lot.

This software will automatically track and summarize each and every bit of information about each vehicle on your car lot. Each vehicle on the lot will be equipped with one of our IntelliTracker chips. These chips are simply plugged into the manufacturer's preinstalled diagnostics port. This hardware, coupled with our management software, will present the manager with the necessary information to make wise economical decisions for the company and sales representatives with the information needed to provide customers with the buying experience that they deserve. This information includes graphs and charts summarizing all activity on a car lot. Better, easier, and smarter – that's IntelliTracker way!

IMF - Vision

The goal of our software (Independent Music Forum) is to facilitate the creation and distribution of independent music by providing the individual pieces of the music industry with a forum through which they can communicate directly with one another. It will also provide a system for funding projects.

Types of utilities that the system provides: UI, Payment, Search, Database, Non-functional, Functional, Communications,

Figure 2. Vision Statements from Class and Special Projects

Summary of Student Experiences:

As the students work on the various stages of the projects they are involved in the following.

- Process of OO Analysis and Design
- Constant need of inter-group communication
- Constant need for intra-group communication
- Periodic preparation and oral presentation of the project
- An enlightened view of the similarity of the Problem Model and the Software Model
- A clear perception of the commonality of elements between projects
- A clear understanding of the process which stems from user requirements and use-cases
- The universality of the OO paradigm in problem solving
- Technical expertise in the OO methodology
- Skills in using UML
- Skills in using an UML tool

They are frequently referred to articles in literature and those available on the web to seek various opinions. Specifically, throughout the implementation of the projects, students take on various roles of:

- Systems Analyst
- Requirements Engineer
- Use-case engineer
- Software analyst
- Design Engineer
- Architect
- Component Engineer

All these activities involve hands-on work and collaborative effort. This is specifically meant to enhance and contribute to their experience in development of a large project. In effect, at the end of the semester, students have gained the following:

- Experience in requirements and use-case engineering
- Experience in the UP process
- Knowledge in OO Analysis and Design
- Skills in the use of UML
- Skills in the use of a Design tool
- Perspective of the universality of the OO approach across domains
- Seamless transition from one OO activity to another.
- Exposure to various technology available for implementation purposes
- An overall perspective of industry strength software and software development process.

Conclusion:

We have presented our method in teaching a course in object oriented analysis and design. Students participate in requirements and use-case engineering as well as a methodical study and use of the UML and UP process and its application for the analysis and design of large-scale application software products. Students undertake multiple of projects, are continuously involved in inter- and intra-group discussions, develop and do multiple oral presentations on software phases, systematically follow the application of the UP process, use an UML and an Design Tool, and study, research, and justify a CASE tool for implementation. Students are exposed to the analysis and design of five different software products using OO methodology and through this they experience and gain insight to the universality and power of OO techniques.

We suggest other institution adopt similar process for their courses in OOA and OOD. We would be glad to work with them and share, and gain, from their experiences.

References:

1. Arlow, Jim and Neustadt Ila, UML and the Unified Process, Addison Wesley.
2. Demuth Birgit, et. Al., OOPSLA98, <http://www.inf.tu-dresden.de/ST2/ST/papers/oopsla98.pdf>