

Object Oriented Software Visual Modeling Course

Using the Unified Modeling Language (UML) & Enterprise Architect (EA) Tool

INTRODUCTION

Course Overview

The Object Oriented Software Visual Modeling (OOSVM) course focuses on UML modeling techniques to conceptualize, develop, realize and transform UML models for requirement gathering and analysis, design, coding, testing and deployment. This course uses a modeling tool, Enterprise Architect (EA), to capture and document models and automate model bi-directional transformation, code generation and re-engineering and documentation generation. Other non-UML models are also covered in this course.

Why Model

Modeling helps visualize, verify, elaborate and document “upstream” software analysis and design activities. “Doing it right early” is key to improve the quality and robustness of software product. The course focuses mainly on upstream activities such as requirements management, requirement analysis and design. Related activities such as coding, testing, round-trip engineering, data modeling, database design and model reviews will also be covered.

Course Structure

This course emphasizes on hands-on practical using real-world project case-study. Participants will be organized into “project” team to solve the case-study problem by applying what they have learnt in this course. At the end of this course, participants will have to produce, present and defend their “project” models.

PRE-REQUISITES

Participants should have the following pre-requisites:

- Some understanding of Object-Oriented (OO) Concepts
- Some experience in OO Programming
- Overview understanding of UML

DURATION: Core Module: 3 days (full-time/part-time) plus optional 1-day per elective module.

VENUE: Our location or in-house

TIME: 9am to 6pm
Core Full-time: 3 Weekdays OR
Core Part-time: 3 Saturdays

Electives are conducted during week-days only.

TARGET AUDIENCE

This course is highly recommended for professionals involved in software requirement definition, analysis, design and implementation such as:

- Systems Analyst, Solution Architect, Software Engineer, Programmer

Professionals who also attended this course in the past:

- Project Manager/Leader, QA Engineers, Software Tester, Technical Writer, CTO

COURSE TOPICS

Part 1: INTRODUCTION

1.1 Software Development Processes & Modeling

Module Objective: At the end of this module, the participants will be able to:

- Appreciate the software development process such as the RUP, CMMI and Agile;
- Understand the relationship between software development process and the need to model; and
- Identify the purposes of UML models.

Module Topics:

- Software Development Process Overview
- Best Practices and Processes: RUP, CMMI, Agile
- Modeling and UML: Why Model, UML introduction
- Overview of 13 UML models: When to use which model and the gaps

1.2 Case Study & EA Briefing

Participants are required to refer to a standard “real-world” case study for all the practical sessions. The case-study describes the requirements of an application to be developed.

Participants are required to use the case study as the basis to apply what they have learnt in this course.

Module Topics:

- Introduction to Case Study and Q&A
- Teaming and Introduction: All practical sessions will be done in groups of not more than 4 participants
- Demo on how to get started with EA

During the practical session, participants are required to:

- Brain-storm in their own group;
- Produce the “project” deliverable (eg. Models) on paper or in EA; and
- Present and defend their work.

80Twenty consultants will critic the deliverable and offer ideas on how to improve or correct the deliverable. Sample solutions will be also provided.

PART 2: REQUIREMENTS MODELING & ANALYSIS

2.1 Requirements Modeling

Module Objective: At the end of this module, the participants will be able to:

- Identify the different types of requirements; and
- Model requirements in EA.

Module Topics:

- Introduction to Requirements Modeling
- Steps in Requirements Modeling
- EA Demo: How to model Requirements?
- Practical: Requirements Modeling

2.2 Use Case Modeling

Module Objective: At the end of this module, the participants will be able to:

- Understand the purposes of Use Case Modeling;
- Identify the structure of a Use Case Model;

- Translate user/business requirements into use case model;
- Identify use cases of a system under development;
- Identify the users and external systems relevant to the system under development;
- Write the use case descriptions; and
- Transform use cases into functional test cases.

Module Topics:

- Introduction to Use Case Modeling: Concepts, principles, scope, purposes and the structure of a use case model
- Steps in developing a use case model:
 - a) Identify Actors and Use Cases,
 - b) Determine Use Case Relationships,
 - c) Determine Use Case Constraints (Pre/Post Conditions) and
 - d) Writing Use Case Scenario Description (Main flow, alternate flow, business rules)
 - e) Transform Use Case into functional test cases
 - f) Establish traceability between Use Cases and Requirements
- EA Demo: How to develop Use Case Model and Document Functional Test Cases?
- Practical: Use Case Modeling

2.3 Domain Class Modeling

Module Objective: At the end of this module, the participants will be able to:

- Identify one of the most important domain/real-world objects of a system;
- Identify relevant objects and filter out the irrelevant ones;
- Identify the attributes of each object;
- Determine the object relationships and multiplicity;
- Transform domain objects into data models; and
- Generate DDL from data model.

Module Topics:

- Introduction to Domain Modeling: Concepts, principles, scope, purposes and the structure of a Domain Model
- Steps to develop and transform domain model:
 - a) Identify domain objects
 - b) Filter irrelevant objects
 - c) Identify the domain attributes
 - d) Determine relationships, multiplicity and roles
 - e) Transform domain model to Data Model
 - f) Generate DDL
- EA Demo: How to develop and transform a domain model?
- Practical: Domain Modeling

- d) Determine the message passing between objects and with itself
 - e) Update Domain model with operations based on the sequence diagram
 - f) Model basic and alternate scenarios using Sequence fragments
- EA Demo: How to develop a Sequence Diagram and Update the Class Diagram?
 - Practical: Sequence Diagramming

3.2 Class Diagramming

Module Objective: At the end of this module, the participants will be able to:

- Understand UML class diagram;
- Finalize the Class Diagrams
- Add detail signatures and other relationships
- Transform Classes to Code and Unit Test

PART 3: DESIGN

3.1 Sequence Diagramming

Module Objective: At the end of this module, the participants will be able to:

- Identify object interactions;
- Identify messages and object life-lines; and
- Model interaction scenarios: sequential, loops, conditionals, alternate/exception, and parallel processing;

Module Topics:

- Introduction to Sequence Diagram: Concepts, principles, scope, purposes and the composition
- Mapping from Requirements and Use Cases to Sequence diagrams
- Steps in developing a Sequence Diagram:
 - a) Identify actors and the 3 major types of objects
 - b) Identify boundary, control and entity classes
 - c) Update domain model with additional boundary and control classes

Module Topics

- Introduction to Class Diagram: Concepts, principles, scope, purposes and the composition of a Class Diagram
- Updating Domain Model to Class Diagram with more details
- Design to Code/Unit Test Code Mapping
- EA Demo: How to add in more details to a Class Diagram and generate code?
- Class Diagramming Practical

PART 4: DESIGN REVIEW AND REFINEMENT

4.1 Design Review and Refinement

Module Objective: At the end of this module, the participants will be able:

- Understand what is a formal process of review;
- Appreciate the need of review;
- Conduct a review of UML models.

Module Topics:

- Peer Reviews Overview and Best Practices

- Review to ensure correctness and completeness of key UML Models:
 - a) Use Case Model
 - b) Domain Model and Class Diagram
 - c) Sequence Diagram
- Practical:
 - a) Participants from each group to review their own practical session solutions to identify any opportunities for improvement.
 - b) Each group to revise their solution accordingly
 - c) Each team is to present their revised solution and highlight their rationales for changes made.

PART 5: FINAL PRESENTATION & DEFENSE

Module Objective: The purpose of this presentation and defense is to assess the participants mastery of the techniques learnt during this course. It is also to simulate the real-world scenario whereby IT professionals have to often present and defend their work/invention during project review meetings.

Each participant will be assessed individually on his/her mastery level in applying concepts learnt. The assessment consists of the following:

- End of practical Quiz (total of 20%): MCQs and short essay
- Final Presentation (80%): The assessment on the participant ability to model, defend own work supported by concepts learnt and the final report submission.

PART 6: ELECTIVE MODULES

The elective modules are optional to pursue. Pick the one or more that are most relevant to your need.

The OOSVM course is the pre-requisite for all the following electives.

6.1 Advance UML Modeling (AUML)

Module Objective: At the end of this 1-day elective module, the participants will be able:

- Model the State Machine and Timing Models

This 1-day elective is recommended to trainees involved in design of real-time, time-critical and/or control systems.

6.2 Enterprise Architect (EA) Tool training (AEAT)

Module Objective: At the end of this 1-day elective module, the participants will be able master the use of EA tool for:

- Automating the software development process;
- Versioning the design repository; and
- Creating a documentation template.

6.3 Enterprise Architect (EA) Tool training for Reverse Engineering (AEAR)

Module Objective: At the end of this 1-day elective module, the participants will be able master the use of EA tool for:

- Reversing programs into design; and
- Reversing databases into design.

This course will cover reverse engineering of Java, C# or C++ programs and relational databases.

COURSE FEES

No	Course Code	Fee (SGD) Per Trainee
1	OOSVM	\$ 1,800
2	AUML	\$ 600
3	AEAT	\$ 600
4	AEAR	\$ 600
TOTAL		\$3,600

Discounts are available for group signup and in-house training.