

Applying UML to Object Oriented Analysis and Design

UML in Context

The Unified Modelling Language™ (UML) is the Object Management Group's (OMG) most-used specification, and is used widely to model not only application structure, behaviour, and architecture, but also business processes and data structure. The OMG has been an international, open membership, not-for-profit computer standards consortium since 1989 and whose board of directors includes representation from almost all organizations that shape enterprise and Internet computing today.

UML stands for Unified Modelling Language and it provides a notation that is readily understandable by all stakeholders of the business, including the business analysts who define the system requirements, the developers who implement the software system and even the client, with a little support, who can readily understand much of the UML model of their proposed software system.

Overview

This course applies the UML to a large, real life case study. The modelling will involve object orientation which will be introduced in easily assimilated chunks throughout the course.

The understanding of the subject matter will be developed through a large amount of practical, hands on work which will be used to consolidate every topic.

We will be looking at different processes that can be used with the UML, but will focus on the Unified Process. This course spends an equal amount of time on both analysis and design and is excellent if you want to put UML and Object Orientation into the context of a full project lifecycle.

Key to becoming an extremely accomplished object orientated designer is knowledge of Design Patterns and how to apply them. This course will

Objectives

By the end of the course attendees will:

- Be familiar with the use of the UML language for describing the artefacts of software and business systems
- Be familiar with one of the major UML CASE Tools.
- Be comfortable with the concepts and philosophy of Object Orientation and be able to contrast it with the Procedural/Functional approach
- Have used the Unified Process
- Be able to create and understand all of the major UML Diagrams
- Be able to construct a domain model as a basis for further design work
- Understand that interaction models are often necessary and be able to use design techniques to construct them
- Be able to use inheritance in design, and understand the purpose of doing so
- Be able to name several of the most useful Design Patterns, and know when they should be applied
- Be able to apply the fundamental design principles

Prerequisites

This course has no prerequisites. All subjects are covered from basic principles; however those with previous programming experience are likely to benefit the most from the Design Patterns section of the course.



UML

Designed and Managed by
the Object Management
Group

Applying UML to Object Oriented Analysis and Design

introduce the idea of Design Patterns and study six design patterns in detail, applying each of them to the case study itself.

Detailed Course Description

Introducing UML

- Models v Diagrams
- What UML is and what it is not
- Modelling Notations
- The Waterfall Lifecycle
- The Unified Process
- The Iterative Lifecycle
- Agile Modelling
- Extreme Programming

Object Orientation Basics

- Pros and Cons of Procedural Programming
- Pros and Cons of Object Orientation
- Classes
- Encapsulation
- Collaboration
- Object to Relational Mapping
- Persistence

Business Modelling

- Beginning the Analysis
- Business Model
- Business Processes
- UML Stereotypes
- Eriksson-Penker extensions
- Business Process Overview Diagram
- Business Actors
- Business Workers
- Case Workers

The Activity Diagram

- Activities, Sub-activities and Actions
- Linking the Activities
- Control Flow
- Object Flow
- Alternative Object Representation
- Initial, Final and Flow Final
- Forks and Joins
- Forks, Joins and Control Flow
- Decision Points and Merges
- Signals
- Objects and Signals
- Swimlanes
- Activity Partitions
- Interruptible Activity Region
- Pins
- Expansion Region
- Parameter Set

Use Cases

- Use Cases as structured requirements
- Granularity of Use Cases
- Uncovering Use Cases from Business Processes
- Primary and Secondary Actors
- Ranking Use Cases
- Specifying Use Cases
- Use Case Descriptions
- Non functional requirements
- Style Guidelines

Applying UML to Object Oriented Analysis and Design

The Domain Model

- Domain Modelling
- Finding Classes
- Attributes
- Associations
- Multiplicities
- CRUD Matrices

The State Machine

- Capturing Business Rules
- Events and States
- Basic Notation
- Superstates and Substates
- Conditional Transitions
- Actions
- Finding Use Cases from the State Model

Detailed Use Case Descriptions

- Use Case Storyboards
- Preconditions
- Postconditions
- Main Flow
- Extension Flow
- Graphical Form

Interaction Modelling - The Communication Diagram

- Objects and Associations
- Method Calls
- Parameters
- Return Values
- Looping
- Creating Objects
- Mapping to the Design Class Diagram

Polymorphism, Inheritance and Composition

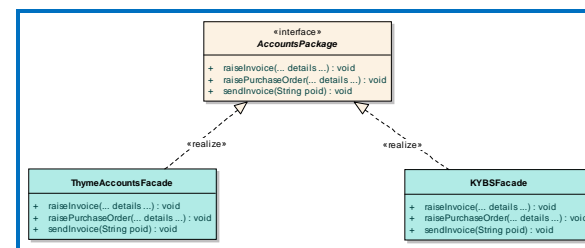
- Inheritance and Overriding in UML
- Abstract Methods and Classes
- Polymorphism
- Interfaces
- Composition and Aggregation
- Design Principle: Program to Interfaces
- Design Principle: Favour Composition

Interaction Modelling - The Sequence Diagram

Design Patterns and Principles

- The 23 Gang of Four (GoF) Patterns
- The Adapter Pattern
- The Façade
- Design Principle: Encapsulate Change
- Design Principle: Least Knowledge
- The Singleton
- Design Principle: Single Responsibility
- Using Factories
- The Strategy Pattern
- The Proxy Pattern

A Resume of all UML 2.0 Diagrams



Object Oriented Analysis and Design in UML

The UML provides a range of both behavioural and structural models for leading and capturing our designs

Applying UML to Object Oriented Analysis and Design

