

ADE200 : UML Overview

Code : ADE200 **Duur :** 1 dag

Categorie : Analysis and Design

Doelgroep :

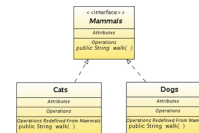
Developers, designers, architecten, managers en andere genteresseerden die een overzicht willen krijgen van de Unified Modeling Language (UML) standaard voor het modelleren van systemen.

Voorkennis :

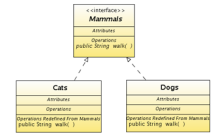
Kennis van en ervaring met systeemontwikkeling en object orintatie is gunstig voor een goede begripsvorming, maar is niet strikt noodzakelijk.

Uitvoering :

De theorie wordt behandeld aan de hand van presentatie slides en wordt afgewisseld met oefeningen. Illustratieve UML modellen worden gebruikt als demonstratie van de concepten.



UML Overview



Inhoud :

Deze cursus geeft een overzicht van de UML taal voor het modelleren van systemen. De UML syntax, symbolen, diagrammen en views worden besproken. Na een inleiding over de UML specificatie en de betekenis van UML als meta taal, wordt in de cursus ingegaan op Structural Modeling en de diagrammen die daarbij worden gebruikt zoals class en object diagrammen en component en deployment diagrammen. Ook domain modellering en het modelleren van de mogelijke relaties tussen de classes zoals inheritance, associaties, aggregaties, composities en dependencies worden besproken. Vervolgens wordt aandacht besteed aan Use Case Modeling, het Use Case Diagram, de rol van de actors en de precieze beschrijving van de stappen in de interactie. Ook het modelleren van het dynamische gedrag van systemen, Dynamic Behavior Modeling, met de bijbehorende interactie diagrammen zoals sequence diagrammen en collaboratie diagrammen komt aan de orde. De modellering van system states middels state charts, het verschil tussen passieve en actieve objecten en de rol van threads zijn eveneens onderdeel van de cursus. En er is aandacht voor de rol van activity diagrammen waarin begrippen als control en data flow en swim lanes worden besproken. Tenslotte staan een aantal geavanceerde UML concepten zoals packages en subsystems, het gebruik van stereotypes, constraints, tagged values en de Object Constraint Language (OCL) op het programma.

Module 1 : UML Introduction

- UML language
- Modeling reasons
- UML Goals
- UML Specification
- UML Building Blocks
- Well Formedness Rules
- Semantic Rules
- Syntactic Rules
- Metamodel Architecture
- UML Metamodel Layer

Module 2 : Structural modeling

- Structural Modeling?
- Core elements
- Core Relationships
- Structural Diagrams
- Classes and Objects
- Types
- Interfaces
- Associations
- Composition
- Generalization
- Dependencies
- Packages
- Links
- Component Diagrams
- Deployment Diagrams
- When to model Structure
- Structural Modeling Tips

Module 3 : Use Case Modeling

- Use Case Modeling?
- Core Elements
- Core Relationships
- Use Case Diagram
- Use Case Relationships
- Actor Relationships
- Use Case Description
- When to model Use Cases
- Use Case Modeling Tips

Module 4 : Interactions

- Interactions
- Core Elements
- Core Relationships
- Instances and Links
- Sequence Diagrams
- Collaboration Diagrams
- Arrow Label
- Kinds of Arrows
- Recursions
- Conditions
- When to model Interactions
- Modeling Tips

Module 5 : State Machines

- State Machine Concepts
- State Machine Diagram
- Diagram Elements
- State Machine Usage
- Event Driven Behavior
- State Machines and Objects
- Object Behavior
- Objects and Threads
- Passive Objects
- Active Objects
- Entry and Exit Actions
- Order of Actions
- Internal Transitions
- State Activities
- Guards
- History States

Module 6 : Activity Graphs

- Activity Diagram Applications
- Control Flow
- Object and Data Flow
- State Machine
- Basis in State Machines
- Steps in Activity Diagrams
- Actions and Sub Activities
- Coordinating Steps
- Decisions
- Sync State
- Convenience Features
- Swimlanes
- When to use Activity Diagrams
- Modeling Tips Activity Diagrams

Module 7 : Advanced Concepts

- Model Management
- Packages
- Core Concepts
- Visibility
- Import and Access
- Package Inheritance
- Subsystems
- Specification and Realization
- Extension Mechanisms
- Semantic Variations in UML
- Stereotypes
- Constraints
- Tagged Values
- Profiles
- Object Constraint Language (OCL)