

UML Diagrams: Concepts and Example Models

1 Introduction

Unified Modeling Language (UML) is a standardized modeling language used to visualize, specify, construct, and document software systems. UML diagrams are divided into structural and behavioral diagrams. This document explains six important UML diagrams along with example models for each.

2 Use Case Diagram

A Use Case Diagram represents system functionality from the user's perspective. It focuses on what the system does rather than how it is implemented.

Key Elements

- Actor
- Use Case
- System Boundary
- Relationships (Include, Extend)

Example Model: Online Shopping System

- **Actors:**
 - Customer
 - Admin
- **Use Cases:**
 - Login
 - Browse Products
 - Place Order
 - Make Payment
 - Manage Products (Admin)

Applications

- Requirement gathering
- Defining system scope

3 Class Diagram

A Class Diagram describes the static structure of a system by showing classes, attributes, methods, and relationships.

Key Elements

- Class name
- Attributes
- Methods
- Relationships

Example Model

Class: User

```
-----  
- userId : int  
- name : string  
- email : string  
-----
```

```
+ login()  
+ logout()
```

Class: Order

```
-----  
- orderId : int  
- date : Date  
-----
```

```
+ placeOrder()  
+ cancelOrder()
```

Relationship: User places Order (Association)

Applications

- Object-oriented design
- Database modeling

4 Sequence Diagram

A Sequence Diagram illustrates how objects interact with each other in a time sequence.

Key Elements

- Objects
- Lifelines
- Messages

Example Model: Order Placement

1. Customer sends `placeOrder()` to `OrderService`
2. `OrderService` sends `processPayment()` to `PaymentGateway`
3. `PaymentGateway` returns payment success
4. `OrderService` confirms order to Customer

Applications

- Understanding control flow
- API interaction design

5 Collaboration Diagram

A Collaboration Diagram focuses on how objects interact and are connected rather than the time sequence.

Key Elements

- Objects
- Links
- Numbered messages

Example Model

- Objects: Customer, `OrderService`, `PaymentGateway`
- Message Flow:
 1. Customer → `OrderService` : `placeOrder()`
 2. `OrderService` → `PaymentGateway` : `processPayment()`
 3. `PaymentGateway` → `OrderService` : `paymentStatus()`

Difference from Sequence Diagram

| Sequence Diagram | Collaboration Diagram |
|-------------------------------|--------------------------------------|
| Time-based Vertical layout | Relationship-based Network layout |

6 Activity Diagram

An Activity Diagram represents the workflow or business process of a system.

Key Elements

- Initial node
- Actions
- Decision nodes
- Final node

Example Model: Login Process

- Start
- Enter Username and Password
- Validate Credentials
- Decision:
 - Valid → Show Dashboard
 - Invalid → Display Error Message
- End

Applications

- Business process modeling
- Algorithm visualization

7 State Machine Diagram

A State Machine Diagram models the lifecycle of an object by showing its states and transitions.

Key Elements

- States
- Events
- Transitions

Example Model: Order Lifecycle

- Created
- Paid
- Shipped
- Delivered
- Cancelled

Transitions:

- Created → Paid (Payment Successful)
- Paid → Shipped (Order Dispatched)
- Shipped → Delivered (Order Received)
- Created/Paid → Cancelled (Order Cancelled)

Applications

- Lifecycle modeling
- Workflow systems

8 Conclusion

Each UML diagram serves a specific purpose in modeling software systems. Including example models improves clarity and understanding of system behavior and structure, making UML an effective communication tool in software engineering.